Using game theory to model interspecific brood parasitism in bird populations.

Martin David Harrison

November 4, 2010

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature:

University of Sussex

Martin David Harrison, DPhil Mathematics

Using game theory to model interspecific brood parasitism in bird populations.

The interaction between hosts and parasites in bird populations has been studied extensively. I use game theoretic methods to model this interaction. This has been done previously but has not been studied taking into account the detailed sequential nature of this game. I introduce models allowing the host and parasite to make a number of decisions which will depend on a number of natural factors. A sequence of events follows, which is broken down into two key stages; firstly the interaction between the host and the parasite adult, and secondly that between the host and the parasite chick. The final decision involves the host choosing whether to raise or abandon the chicks that are in the nest. There are certain natural parameters and probabilities which are central to these various decisions; in particular the host is generally uncertain whether parasitism has taken place, but can assess the likelihood of parasitism based upon certain cues (e.g. how many eggs remain in its nest).

I have taken elements of games which have been previously created and constructed my own models to fully describe this interaction. These parasites have different methods of parasitizing the nests of their hosts, and the hosts can in turn have different reactions to these parasites. This is later built into a model where there is more than one host nesting over a breeding season. We have a number of nesting sites and different time points in which the host can begin to nest. In the previous models the host was given the opportunity to abandon the nest. In this game the host is allowed to abandon and then restart the nesting process. The probability that the host is parasitized can be decided using a number of factors including the number of hosts laying during a given time period, the nesting site or the number of parasites during the course of the season.

Using these models we are able to find situations which match those which we have seen in nature. Also the models are able to predict what natural changes such as parasitism rate or mimicry will do to the interaction. Overall I believe these models to give as good an indication of the key elements of the interaction and how they can change over time.

Acknowledgment

I would like to thank EPSRC for the funding of this project. I would also like to thank the University of Sussex for use of its facilities both on campus and online. I would like to thank Dr Mark Broom for all his help in being my DPhil supervisor. Thank you to my family in particular my Mum, Dad, Nan and Grandad for all their support. I would also like to thank members of the Brighton Tsunami American Football Club especially Antony Bainbridge and Marcus Dilley for some technical support provided.

Figure 1: Photograph of Reed Warbler feeding a Cuckoo Chick

Contents

Chapter 1

Introduction

1.1 Brood Parasitism

Many species of bird parasitise others by laying their eggs in their nests (brood parasitism, e.g. Payne [39]). It involves the introduction of an egg into a previously laid 'host' nest by a parasite. Sometimes such parasitism occurs within species (intraspecific) and sometimes the victims are other species (interspecific). Typically intraspecific parasites also form their own nest, but interspecific parasites do not, and are thus completely reliant on their hosts to raise their offspring; they are referred to as obligate brood parasites [9]. There are six clades of birds which exploit the post-hatching care of other species; the Old World Cuckoos, the Clamator Cuckoos, the New World Cuckoos, the Honeyguides, the Vidua Finches and the Cuckoo-Finch (*Anomalospiza Imberbis*), and also five species of Cowbird [49]. The reproductive biology of the brood parasites is broadly similar between species, but the behaviour of their chicks differs in one key aspect. Soon after hatching some parasite chicks (from the Old World Cuckoos, some of the New World Cuckoos, the Cuckoo-Finch and the Honeyguides) deliberately kill the host young, either by evicting them from the nest or by using their hooked bills to inflict lethal injuries. The remaining species do not do this, and generally at least one of their companions in the nest survives to fledge.

What is the cause of such differences in behaviour? One possibility is that species that do not kill host young either suffer from evolutionary lag or are not physically capable due to the relatively large size of host young. There is some strong evidence for evolutionary lag since the most recently evolved brood parasites tend to be those that do not kill chicks. However, there are exceptions. For example, the Brown-headed Cowbird (*Molothrus Ater*) at times strategically evicts host offspring

from the nest [11], and two Old World Cuckoo species appear to have lost the capacity to kill young.

An alternative explanation for the difference is to consider that the possible costs that parasitic offspring may experience when they kill nest-mates might limit the evolution of host-killing [21]; for instance such a cost is an increased risk of desertion by the host parents [25]. In our first model we shall consider a single interaction between a host and its parasite, which will involve potential strategic choices at different stages.

Several decisions can be made by the adult host and parasite and also by the parasite chick once it has hatched. These decisions include (for the host) ejection of the parasite egg [39, 28], abandonment of the nest [47], or to continue to raise the clutch with the parasite intact [27]. The adult parasite can decide to eject some of the host eggs whilst it lays a parasitic egg [9] or can just add the egg but otherwise leave the nest as it is [27].

There is a cost to the host in raising a parasite chick [16], whether this be in the destruction by the parasite of its own chicks it has spent time to raise or in the increased cost of raising the parasite chick [23]. There are also costs in trying to resist parasitism [8]. For example, some parasites have evolved to the point where they are able to mimic the host egg to a good degree [50]; this can cause the host to eject the wrong egg. There is also the possibility that it could also damage its own nest in trying to eject the parasite [17]. Thus the host must balance the costs of resisting this parasitism with the potential benefits of resistance, the cost-benefit equilibrium [56].

1.1.1 Egg Destruction

There is much documented and even video evidence of cuckoos ejecting host eggs and even chicks. N.B. Davies book 'Cuckoos, Cowbirds and Other Cheats' [9] and Ian Wyllie's book 'The Cuckoo' [57] both describe this behaviour. Common Cuckoos (for example) observe nesting sites being built or at least materials being taken to a site. Once they find the nest they lay one of their own eggs and remove one of the host eggs. The time it takes to lay an egg and remove the host egg is very quick (within a few seconds).

It has been known for many years that cuckoos are reared alone. But it was not until Edward Jenner (the inventor of vaccination) [57] observed the hatched cuckoo chick ejecting the other host eggs and chicks that we knew how this occurred. The hatched cuckoo uses a special call equivalent to that of many host chicks in order to obtain more food from the adult host to grow to the size

it needs to be. Cowbirds however, although being parasites, do not show the ejection behaviour of the cuckoos.

1.1.2 Egg Rejection

As a counter to egg destruction, some hosts have built a defence mechanism. Biological studies ([38], [28], [34],[35]) have shown hosts rejecting eggs. The methods in which an egg is ejected can be different for each host species. Yellow Warblers reject by burial of the parasite egg in the nesting material [46], other species puncture the eggs [27], some grasp and throw the eggs from the nest [36] and others merely abandon the nest [47]. There is a possibility that in trying one of these it will damage its own brood (which must happen in the case of abandonment) [17].

1.1.3 Egg Mimicry

As a counter to egg rejection, some species of cuckoos have developed egg mimicry, that is making its eggs look like those of the host. When this occurs it means the parasite can only parasitise one species of host and this complicates the egg rejection decision by the host. Even though some mimicking species, for example the common cuckoo, can parasitise many hosts [6]. If the mimicry is good then it could eject one of it's own, Payne [38] has shown that this can occur. Brown-Headed Cowbirds do not show mimicry, this could be because they need to parasitise different hosts or the hosts do not show an ejection behaviour.

1.1.4 Nesting Site Choice

There are several factors that affect nesting site choice for host species. The interesting factor for us is the impact parasitism has on nesting site choice. One aspect discussed by Clarke et al [5] is the independent variable 'Cuckoo View', meaning how direct the cuckoo's view of a particular nest is from its vantage point in a tree. Their model showed three significant contributors to parasitism rates; site, cuckoo view and neighbourhood view (that being the view of the whole neighbourhood of nests). The results showed a significant increase in parasitism as the nesting site view became more direct.

Øien et al [35] have worked on models of parasitism of warblers with respect to the distance of the nest, situated in reeds near a lake or river, from the nearest tree. They showed overall that those who lay away from trees lay earlier in the year, lay larger eggs and have a lower parasitism

rate.

1.2 Previous Models

Significant mathematical modeling work in the brood parasitism field has been done by Takasu et al [51, 52, 53]. Much of his work considered the dynamics of a whole population of hosts and/or parasites, focusing on the underlying genetics and the co-evolution between the host and parasite in the form of an arms race describing the adaptation of the level of rejection and mimicry behaviour over time. Evolution typically occurs in the following stages. Firstly hosts neither recognize nor reject parasites and there is no mimicry. Then hosts establish defences against eggs that look different, since there is no mimicry the parasites may become extinct. Finally parasites evolve better mimicry forcing the host to raise rejection levels or give up rejection completely due to the associated costs. Takasu [51, 52, 53] considers the possible outcomes from this co-evolutionary process in parasite and host behaviour, and in egg appearance. He also looks into the evolution of the host-parasite interaction over a succession of breeding seasons, as opposed to just one interaction or even one single breeding season. Yamauchi [58] develops an evolutionary model to describe the conditions under where intra- and interspecfic nest parasitism occurs. His model predicts that the origin is probably in intraspecfic parasitism but under certain conditions, such as a reduction the in cost of doing so, interspecfic nest parasitism can evolve. He also predicts that under a low marginal decline in survival rate of the offspring conspecific (laying in both own and other species nests) brood parasitism can evolve.

Previous models of this behaviour have used game theoretical methods [4, 7, 30, 42], for example Pagel et al [37] have provided a model of the evolution of ejector and non-ejector host birds, mostly in relation to cuckoos. Rodríguez-Gironés and Lotem [43] and Lawes and Marthews [26] discuss the egg rejection problem with regards to parasitism rate and egg mimicry. Zink [59] has modelled the behaviour of intraspecfic brood parasitism, looking at when this is beneficial to co-operative or solitary breeding. Schmidt and Whelan [45] discuss the impact of nest predation and brood parasitism and what level of modelled defence should be allocated to each. We shall consider some of these models in more detail.

1.2.1 Servedio-Hauber Model

Introduction

The Servedio-Hauber Model [47] looks at the question as to whether the host should eject or abandon if it believes there to be a parasite amongst its brood.

This model is an attempt to explain the different rejection behaviours shown by hosts within species when faced with different types of parasite.

Essentially the host will have one of three choices

1. Eject
2. Abandon
3. Do Nothing

Doing nothing is most likely a response of the host either not being able to perform the other tasks or not willing to risk the costs of the other choices.

Nest abandonment is the only possible defence against parasitism for some of the hosts who cannot grasp, eject or otherwise destroy the parasite egg. Nest abandonment can also incur temporal, energetic and social costs associated with finding a suitable site for re-nesting. However both forms of anti-parasite behaviour can be seen. It can be shown that the decision as to which strategy to use amongst hosts is not random. Most cuckoo hosts eject eggs [25] and those who are hosts of Cowbirds tend to abandon the parasitised nests [19].

One explanation of this is that the hosts of cowbirds have not been parasitised as long, from an evolutionary perspective, as those who host cuckoo nests and therefore have not gained the ability to effectively discriminate and eject cowbird eggs [6]. However it has also been shown that some cuckoo hosts abandon [33] and some Cowbird hosts do possess the ability to eject eggs[44]. This model therefore looks at the relative cost and benefits of both strategies of defending against brood parasitism.

Method

To understand the fitness of host strategies the authors take an average over the whole population rather than looking at parasitised versus non-parasitised nests. The goal is to calculate the average fitness of those hosts who respond to parasitism by nest abandonment as opposed to those who

respond by egg ejection. Both are assumed to be parasitised at random with probability p_1, with parasitic birds removing an egg before laying *one* of their own. The hosts then use a decision rule based upon an arbitrary linear trait difference between the parasite and host eggs.

This is then used to decide whether an egg is unacceptably different to that of a normal host egg. If this is found to be so then egg ejectors eject said egg and continue to raise the brood whereas nest abandoners attempt to re-nest. The re-nesting then has a probability of p_2 of being parasitised, with each egg having a lower relative fitness than those in the first nest. For algebraic simplicity the second nest is assumed to be raised whether or not there is a parasite present in the nest.

Parasites are then assumed to be *cowbird-like*, in that the parasite is raised with the brood. With this, the host will have a reduced relative fitness compared with that of a nest which contains no parasite. If a nest is parasitised successfully by a *cuckoo-like* parasite however, the clutch will be completely destroyed leaving the host with no fitness whatsoever.

Results

Under most parameter values, for hosts of *cowbird-like* parasites, egg ejection was found to be the better option. However some hosts, due to some variations in life history traits that vary across different species and across populations of hosts, have switch points where nest abandonment yielded a higher fitness than that of the egg ejection strategy. These include rates of parasitism, fitness rates of re-nesting attempts compared to first attempts and higher fitness values of eggs in parasitised nests also led to switching points between egg ejection and nest abandonment. Under *cuckoo-like* parasitism egg ejection always has a higher fitness pay-off than that of nest abandonment because of the assumption that the parasite is raised with no host chicks.

1.2.2 Broom-Ruxton-Kilner Model

Introduction

The idea behind this model [4] is that brood parasites vary in the amount of harm they inflict on their hosts. The authors use a game-theoretic model to attempt to account for this variation. Unlike previous models this one considers the fact that hosts may abandon a single nestling regardless of whether it is a host or a parasite. The parasite chick must decide whether or not to kill all the other host young by balancing the gains it stands to make by lack of competition against the possibility of the host abandoning the nest.

Method

First of all the following sequence of events is assumed to happen:

1. The host forms a nest and lays a clutch of eggs.
2. Since the parasite destroys down to one egg we need to look at the other ways it can be destroyed to one egg. The number of eggs can be reduced to one by either of two events;
   - a parasite egg is laid in the nest (with probability P) and the parasite destroys all the host eggs (with probability C_P)
   - there is destruction by natural causes down to one host egg, which happens with probability E.

   It is possible for both events to occur at the same time but this is assumed to occur sufficiently rarely that it has been omitted from this model. Thus we have the following probabilities
   - the probability the brood will be reduced to a single individual ($E + PC_P$);
   - the probability the brood will increase by a single individual ($P(1 - C_P)$);
   - the probability the brood will remain the same ($1 - P - E$).

   Some parasite birds lay an egg whilst also destroying one of the host's, but this complication is unnecessary since they do not assume a discriminatory behaviour for the host other than being able to differentiate between one and several.

Probability	Description	Host Payoff	Parasite Payoff	Probability of host adult survival
$1-P-E$	No parasitism occurs	B	NA	μ_b
$P(1-C_P)$	Parasitism occurs without ejection	cB	θ	μ_{bp}
EC_H	A lone host chick is deserted	0	NA	μ_d
$E(1-C_H)$	A lone host chick is not deserted	1	NA	μ_s
PC_PC_H	A lone parasite chick is deserted	0	0	μ_d
$PC_P(1-C_H)$	A lone parasite chick is not deserted	0	1	μ_p

Table 1.1: Table of payoffs for hosts and parasite showing where parasitism occurs in the Broom-Ruxton-Kilner Model

3. The host must choose to desert or not; it is assumed that the host cannot detect parasite eggs in its nest, it can only see if there is *one* egg or *several*. If there are *several* then the host will always raise. If there is one then the host deserts with probability C_H and does not desert with probability $1-C_H$.

If the parasite is abandoned it receives no reward. If it is raised on its own then it receives a reward of 1. If it is raised along with other host chicks it receives a reward θ which is assumed to satisfy $0<\theta<1$. Also defined are $0<c<1$ as the value of the brood chicks brought up with the parasite and B as the value of the full brood relative to the value of a single host chick.

This gives us the possibilities of host adult surviving over winter shown in Table 1.1 where $\mu_{bp} \leq \mu_b \leq \mu_s \leq \mu_d$ and $\mu_{bp} \leq \mu_p \leq \mu_s \leq \mu_d$. For instance it is more likely to survive after abandoning μ_d than if it raised a single chick μ_s. To determine ESSs we must find an equilibrium pair (C_h, C_p) where each party cannot increase or decrease and achieve a increased reward. We set up the following.

- $R(C_h, C_p)$ as the reward function to the host under host strategy C_h and parasite strategy C_p.
- $S(C_p, C_h)$ as the reward function to the parasite under host strategy C_h and parasite strategy C_p.

From this we get that

$$S(C_p, C_h) = \theta(1-C_p) + C_p(1-C_h) = \theta + C_p(1-\theta-C_h) \tag{1.1}$$

and

$$R(C_h, C_p) = \frac{(1-P-E)B + PcB + E - PcBC_p - EC_h}{1-\mu_b + P(\mu_b - \mu_{bp}) + E(\mu_b - \mu_s) + PC_p(\mu_{bp} - \mu_p) + EC_b(\mu_s - \mu_d) + PC_pC_h(\mu_p - \mu_d))} \tag{1.2}$$

in order to find an equilibrium pair (C_h^*, C_p^*) we must find C_h^* which maxmises $R(C_h^*, C_p)$ and C_p^* which maxmises $S(C_p^*, C_h)$.

Determining Evolutionarily Stable Strategies

If we define

$$\alpha = (1-P-E)B + PcB + E - PcBC_p^*,$$

$$\beta = -E,$$

$$\gamma = +P(\mu_{bp} - \mu_p)C_P^*$$

and

$$\delta = E(\mu_s - \mu_d) + PC_p^*(\mu_p - \mu_d)$$

with

$$G_0 = E(-1(1-\mu_b) + P(\mu_{bp} - \mu_b) + E(\mu_d - \mu_b) + B(\mu_d - \mu_s)(1-P-E+cP)),$$

$$G_1 = (\mu_d - \mu_{bp})EP + (\mu_s - \mu_d)PcBE + (\mu_d - \mu_p)PB(1-P-E+cP),$$

and

$$G_2 = P^2cB(\mu_p - \mu_d).$$

The possible ESS solutions include the following

- The parasite always evicts host eggs and the host never abandons

$$(C_p^* = 1, C_h^* = 0).$$

This occurs when $\beta\gamma - \alpha\delta < 0$ i.e $G_0 + G_1 + G_2 < 0$.

- The parasite never evicts the host and the host always abandons a lone egg

$$(C_p^* = 0, C_h^* = 1).$$

  This occurs when $\beta\gamma - \alpha\delta > 0$ i.e $G_0 > 0$.

Both of these are sensible and match what we find in nature. For those solutions where C_p^* is either 1 or 0, but C_h^* is neither 0 nor 1 we require certain combinations of the Gs. Since it is highly improbable that these combinations take these values these are omitted.

We need solutions for C_h^* when C_p^* is neither zero nor one. For this we need

$$C_h^* = 1 - \theta.$$

This can only occur if $\beta\gamma - \alpha\delta = 0$. That is, solutions of the form $(0 < C_p^* < 1; C_h^* = 1 - \theta)$ can only occur at the roots of

$$G_0 + G_1 C_p^* + G_2 (C_P^*)^2 = 0.$$

We have the following five conditions:

- (A) $G_1^2 > 4G_0G_2$;
- (B) $G_0 > 0$;
- (C) $G_1 > 0$;
- (D) $G_1 + 2G_2 > 0$;
- (E) $G_0 + G_1 + G_2 > 0$.

If (A) is not satisfied then $C_P^* = 1$ is the unique ESS otherwise we get those described in Table 1.2.

This model goes a long way to describing the interaction between host and parasite and also gives solutions which are shown to appear in nature. It shows occasions where populations of hosts and parasites can be stable.

There are also some numerical results which show the effect of the change of some of these parameters. These include

	B	C	D	E	
1	Y	Y	Y	Y	$C_p^* = 0$ is the unique ESS
2	Y	Y	Y	N	This combination is impossible to achieve
3	Y	Y	N	Y	$C_p^* = 0$ is the unique ESS
4	Y	Y	N	N	Both $C_p^* = 0$ and 1 are ESS and intermediate unstable equilibrium
5	Y	N	Y	Y	This combination is impossible to achieve
6	Y	N	Y	N	This combination is impossible to achieve
7	Y	N	N	Y	$C_p^* = 0$ is the unique ESS
8	Y	N	N	N	Both $C_p^* = 0$ and 1 are ESS and intermediate unstable equilibrium
9	N	Y	Y	Y	There is a stable intermediate equilibrium only
10	N	Y	Y	N	$C_p^* = 1$ is the unique ESS
11	N	Y	N	Y	There is a stable intermediate equilibrium only
12	N	Y	N	N	$C_p^* = 1$ is an unique ESS + 2 intermediate equilibria, one stable, one not
13	N	N	Y	Y	This combination is impossible to achieve
14	N	N	Y	N	This combination is impossible to achieve
15	N	N	N	Y	This combination is impossible to achieve
16	N	N	N	N	$C_p^* = 1$ is the unique ESS

Table 1.2: Where (A) is met, there are 16 different possible combinations for the four conditions either to be met (denoted Y) or not met (denoted N). These are described along with the possible solutions that each allows.

1. As the value of a full nest (B) increases, the probability of abandonment by the host increases due to the probability of eviction by the parasite increasing.

2. As the effect on the host brood by the parasite increases the parasites tends to eject less often.

3. The probability of parasitism has no effect on the ejection and abandonment strategies.

We shall take many of the methods used in this interaction such as finding equilibrium pairs but take this into a more individualised model. We also notice the last point here is that the probability of parasitism has no effect on the strategies. This is something that has not been shown in other models described earlier. This is due to the fact that this is a constant over time in this model and the parasitism this year is going to be the same next year so abandonment is not necessary. We shall look to see if we can make any predictions of the change in parasitism rate and what effect that will have on the interaction.

1.2.3 Takasu's Models

Fugo Takasu has done a multitude of work ([51, 52, 53]) in the brood parasitism field focusing on the mathematical modeling of these interactions. As stated earlier in the chapter many of his papers look into the adaptive dynamics of a whole population of hosts and/or parasites. He explores the genetic makeup of alleles as a way to construct models of rejecters and acceptors in host populations. In most of his work he describes allele A and allele R where R causes rejection and A causes acceptance, with allele R being dominant over allele A. He describes x_t, y_t and z_t as the frequency of RR, RA and AA genotypes in the host population in year t, with these summing to one.

Takasu's Cuckoo Host Association Model [53]

This model looks at the evolution of defence mechanisms by the host, specifically finding evolutionarily stable strategies that specify populations of rejecter and acceptor hosts against a cuckoo-type parasite, where the host accepting a cuckoo always results in total destruction of the host's offspring but rejecters having a cost to pay in rejecting, be this an ejection or recognition cost.

Let P_t be the population density of the female cuckoo and H_t be that of the female host in year t. A female cuckoo will lay one egg in the host's nest if she finds the nest, and with probability Γ a female cuckoo chick grows up from a parasitized nest and survives to the next breeding season. The adult female cuckoo survives to the next breeding season with probability s_p. Assume that the host response to parasitism is determined by the two alleles, allele A and allele R. In the absence of a cuckoo the acceptor pair raises f female offspring, while the rejecter pair raises εf female offspring, which is less by a factor of ε. This parameter measures the cost of rejection. It's value is generally less than, but close to, one.

To model the parasitic behaviour of the cuckoo, Takasu assumes that the female searches independently and randomly with a searching efficiency measured by parameter a. The probability that a host nest escapes parasitism is given by the zero term of a Poisson distribution, e^{-aP_t}. The value of parameter a will vary from host species to host species. Under these assumptions, the number of female offspring from an acceptor pair is fe^{-aP_t}, while the number from the rejecter pair remains the same. This can be seen in Figure 1.1.

Cuckoo parasitism is successful in the nest of an acceptor pair, of which the frequency is z_t^2. Thus the population density of the cuckoo in the next year is:

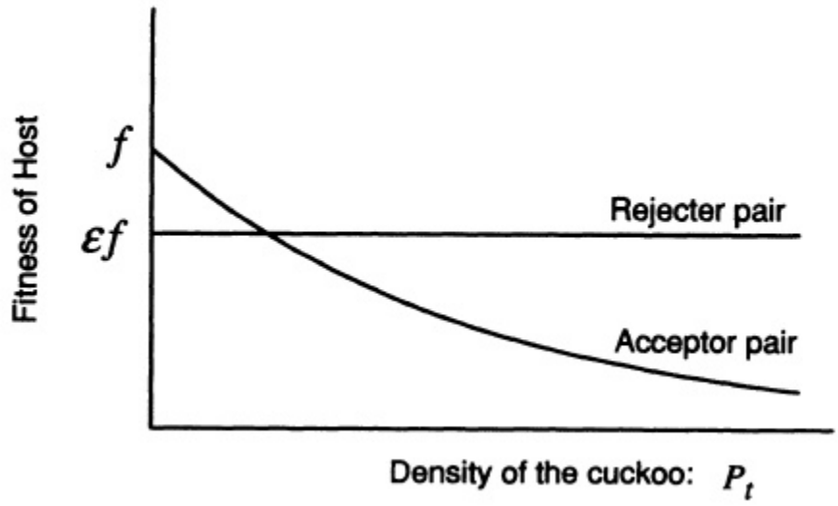


Figure 1.1: Fitness of Rejector and Acceptor Hosts for different density of cuckoos' population in Takasu's association Model [53]

$$P_{t+1} = s_p P_t + (1 - e^{-aP_t}) z_t^2 H_t \Gamma,$$

where the first term of the right side represents the density of adult survivors and the second term represents the newly acquired yearlings. Therefore the parasitism rate is:

$$(1 - e^{-aP_t}) z_t^2$$

Next, he considers the dynamics of the host population. The density of rejecter pairs is $(1 - z_t^2)H_t$, each of which produces εf offspring and $z_t^2 H_t$ acceptors which produce $f e^{-aP_t}$. Thus the total density for the offspring is:

$$[(1 - z_t^2)\varepsilon + z_t^2 e^{-aP_t}] f H_t,$$

The host density in the absence of the cuckoo is generally regulated by the availability of limited food or territories. Taking into account this effect, we have the following equation for the host density in the next year

$$H_{t+1} = \frac{1}{1 + H_t/k} \{ s_H H_t + [(1 - z_t^2)\varepsilon + z_t^2 e^{-aP_t}] f H_t \},$$

where s_H is the survival rate of the adult host and the factor $\frac{1}{1+H_t/k}$ represents the density

effect of intraspecific competition on the host population. The scale of this effect is measured by $1/k$.

The densities of offspring with genotypes RR, RA and AA are derived as follows:

- RR - $(x_t^2 + x_t y_t + y_t^2/4)\varepsilon f H_t$
- RA - $(x_t y_t + 2x_t z_t + y_t z_t + y_t^2/2)\varepsilon f H_t$
- AA - $(y_t^2/4 + y_t z_t)\varepsilon f H_t + z_t^2 f exp(-aP_t)H_t)$

From this he works out the frequencies of the alleles RA and RR for the following year, given by

$$x_{t+1} = \frac{s_H x_t + (x_t^2 + x_t y_t + y_t^2/4)\varepsilon f}{s_H + (1 - z_t^2)\varepsilon f + z_t^2 f e^{-aP_t}}$$

and

$$y_{t+1} = \frac{s_H y_t + (x_t y_t + 2x_t z_t + y_t z_t + y_i^2/4)\varepsilon f H_t}{s_H + (1 - z_t^2)\varepsilon f + z_t^2 f e^{-aP_t}}.$$

We can find equilibria for the parasite population P, host population and density of the alleles by setting $P_t = P_{t+1} = P^*$, $H_t = H_{t+1} = H^*$, $x_t = x_{t+1} = x^*$ and $y_t = y_{t+1} = y^*$. From this we get an ESS of

$$(P^*, H^*, x^*, y^*) = \begin{cases} (0, k(f + s_H - 1), 0, 0) \\ (\tilde{P}, \tilde{H}, 0, 0) \\ (0, k(\varepsilon f + s_H - 1), 1, 0) \\ \left(\frac{1}{a} log \frac{1}{\varepsilon}, k(f + s_H - 1), (1 - D^{1/4})^2, 2(D^{1/4} - D^{1/2})\right) \end{cases}$$

where $\tilde{P}$ and $\tilde{H}$ is the unique solution of

$$\tilde{H} = \frac{(1 - s_P)\tilde{P}}{\Gamma(1 - e^{-a\tilde{P}})},$$

$$\tilde{H} = k(fe^{-a\tilde{P}} + S_H - 1)$$

and

$$D = \frac{(1 - S_P) log\left(\frac{1}{\varepsilon}\right)}{ak\Gamma(\varepsilon f + S_H - 1)(1 - \varepsilon)}$$

From this Takasu was able to find an evolutionarily stable strategy for the population of hosts and parasites under the given conditions, giving situations where both hosts and parasites can co-evolve with populations of both staying the same. The solution $(\tilde{P}, \tilde{H}, 0, 0)$ shows the situation where only acceptor pairs will survive but the solution $\left(\frac{1}{a} log \frac{1}{\varepsilon}, k(f + s_H - 1), (1 - D^{1/4})^2, 2(D^{1/4} - D^{1/2})\right)$ gives a stable strategy where hosts of all types can survive. This has a similar philosophy to the Broom-Ruxton-Kilner model but takes a different approach using the pairs as the evolutionary factor. Both have merits and show strategies where rejecting and accepting can evolve.

A theoretical consideration on co-evolutionary interactions between avian brood parasites and their hosts [52]

This model looks into the adaptations and counter-adaptations of host and parasite in the interactions between them. These include the hosts being aggressive towards parasites near the nesting sites, recognition and rejection of parasite eggs in nests and parasite mimicry. The evolutionary process can be viewed in terms of an *adaptive* landscape depicting fitness plotted against a continuous set of possible strategies. The strategy in this cas-e is the level of rejecter behaviour r, where $0 \leq r \leq 1$, in which $r = 0$ corresponds to no defence and $r = 1$ to perfect defence.

If the host's defence entails a small cost, such as recognition or rejection errors, then the mean fitness should peak at some fixed $r = r' < 1$. This is because there is a perfect defence at $r = 1$, so no parasites can reproduce and therefore there is no risk of parasitism. Therefore there is no merit in discriminating and rejecting parasitism given the cost involved. If there is no cost to mimicry however then the adaptive landscape is an increasing function and will level off at $r = 1$.

The other evolutionary adaptive landscape is that of the parasite's mimicry. Let m be the level of mimicry of the egg ($m = 0$ being very poor mimicry and $m = 1$ being perfect mimicry). This usually results in an increasing function of parasite fitness as better mimicry results in more successful parasitism. However the level of mimicry depends on the host's strategy r. If the host shows no defence (i.e. $r = 0$) then mimicry becomes unnecessary. As r increases then the level of mimicry should increase with it. The co-evolutionary arms race is one that describes the adaptation of these levels of rejecter and mimic behaviour over time and can be in different time scales, so that one adapts faster than the other. The general consensus [6, 9, 44] on how this evolves is in

the following stages:

1. Hosts neither recognize nor reject parasites and there is no mimicry.

2. Hosts establish defences against eggs that look different. Since there is no mimicry the parasites may become extinct.

3. Parasites evolve better mimicry forcing the host to raise rejection levels or give it up completely due to the cost.

Takasu comes to the conclusion that there are occasions where the arms race can be evolutionarily stable [52, 51]. These usually fall into one of the three categories shown above. We shall see if, in our model, we can find situations where these occur.

We summarise the models and their findings into Table 1.3. Many of these consider the evolution of interspecific brood parasitism from intraspecific in terms of a whole population. Servedio and Hauber [47], Maruyama and Seno [30] and Robert and Sorci [42] look at the individual interaction of the host and parasite. Many of these models consider the co-evolution of the defences such as egg recognition and parasite mimicry.

None of these models have taken into account the whole game and how all these factors effect the individual interactions between host and parasite. We shall begin to factor in these elements to develop an all encompassing model.

Table 1.3: Summary Table of Past Models

Model	Reference	Description	Key findings
Servedio and Hauber	[47]	Looking at different defenses against parasitism	Suggested different situations where abandoning or ejection can happen.
Broom et al	[4]	Looking at the abandoning/not abandon behaviour of hosts and the destroy/not destroy behaviour of the hosts.	Produces a model under predicts under which conditions parasites of the type cowbird and cuckoo can evolve
Takasu et al	[51, 52, 53]	Modelling adaptive population dynamics using reject and eject genotypes	Finds ESSs where populations of reject and accept can survive alongside different parasite types.
Yamauchi	[58]	Looking at the evolution of intraspecfic and interspecific brood parasitism.	Suggests a beginning in intraspecfic brood parasitism but moving on to other types under certain conditions
Davies	[7]	Looking at the additional factors of recognition errors and probability of parasitism in order to determine accept/reject behaviour. Uses game theoretic techniques	Found that under lower parasitism rates accepting parasites was a better idea because of the chance of recognition errors.
Maruyama and Seno	[30]	Discussing the problem of how many eggs the parasite should lay in the host's nest.	Was found that the number of eggs laid by the parasite significantly affects the success of the parasite.
Robert and Sorci	[42]	Present a simple analytical model to investigate the conditions for the evolution of interspecific brood parasitism	They produced model finding under which conditions it is optimal to either lay $n > 0$ or $n = 0$ of its own eggs. The latter meaning it is an interspecific brood parasite.
Rodríguez-Gironés and Lotem	[43]	Looked at a 'Signal-Detection Theory' model for recognition of parasite eggs.	Shows conditions when detection is useful but, when mimicry is good and individual variability is relatively high, females must use what they refer to as an extended learning phase.
Lawes and Marthews	[26]	Look at a model of parasite recognition with regards to parasite population.	Gives conditions under which the parasite should evolve recognition as a defense
Zink	[59]	Produces a model to find ESS where intraspecific brood parasitism (IBP) is preferred over cooperation and solitary breeding.	This model makes several predictions regarding the conditions under which IBP may evolve, including the parasites being able to produce more young, and constraints on the solitary hosts.
Schmidt and Whelan	[45]	A model looking at the impact of predation and parasitism	They suggest that in the majority of the cases developing a defense against predation is as important as parasitism.

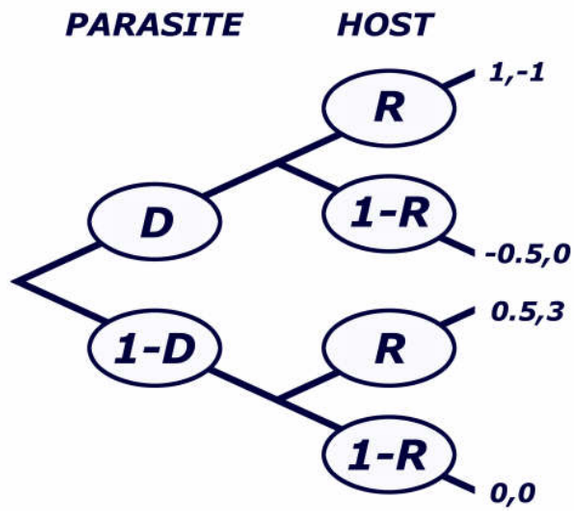


Figure 1.2: Decision Tree with the parasite making a decision followed by the host making a decision based upon what the parasite has done previously. D is the proportion of time the parasite chooses to *Destroy* and R is the probability that the host chooses to *Raise*. The numbers at the end of the tree represent the reward to the parasite and reward to the host respectively.

1.3 Extensive Form Games

The idea of this section is to give an insight into extensive form games as well as some of the factors taken from previous models that can be modeled. We look at some simple numerical examples to give a flavour of the more analytical work to come.

1.3.1 Game 1 - Perfect Information

While previous models of this interaction have been created, none have really taken into account the sequential nature of the game. In order to study this interaction we must look at the possibilities with regard to transfer of information. We set up the game as in Figure 1.2. From this we can see that there are four possible outcomes for this game.

- Parasite destroys all host eggs and host raises the parasite chick.
- Parasite destroys all host eggs and host does not raise the parasite chick.
- Parasite does not destroy and host raises the full brood.
- Parasite does not destroy and the host does not raise the brood.

We assume there is some parasitism occurring in this example and we set the game up so that the host simply receives no reward if it chooses not to raise, receives a reward of 3 if it raises an undestroyed nest (4 for its own egg minus 1 for the parasite egg) and gets -1 if it raises just a parasite egg. The parasite receives a score of 1 if it is raised alone, 1/2 if it is raised with other host eggs, 0 if it does not destroy and is not raised and a penalty of 0.5 if it is not raised after destroying.

We solve these kinds of games by backwards induction. We look at what the host will do in given all the choices for the parasite and then let the parasite choose which decision works out best for him given what the host will do afterwards.

If we assume *Perfect information* (that the host knows whether the parasite has destroyed or not) then the only equilibrium is (*Not Destroy*, *Raise*) because if the parasite plays *Destroy*, the host will play *Not Raise* whereas if the parasite plays *Not Destroy* the host will play *Raise*. Since the bigger reward for the parasite is the second of those two scenarios, that is our equilibrium. However we see destroyed nests being raised by cuckoo hosts, so this cannot be the story in the real world.

1.3.2 Game 2 - Incomplete Information

Thus we must rethink the situation. One could argue that the game has imperfect information, that the host does not know which choice the parasite has made. However this does not make much sense as the host should be able to tell the difference between a full clutch of eggs and just one egg. This means that we must see this game as a problem of incomplete information i.e. there is some other factor acting on the game that is not caused by the host or parasite. This will be something that happens naturally (i.e. not a decision made by the host or parasite). We set the game up as shown in Figure 1.3.

The additional element of this is the probability of parasitism occurring in the first place with p being the probability of parasitism. If there is no parasitism the host will receive a value of 4. So the host must now choose *Raise* or *Not Raise* based upon this probability and the information on the parasite.

In this instance if the parasite chooses *Not Destroy* the best option is still to choose *Raise* as it is the best outcome no matter the value of p. However if the parasite chooses *Destroy* the host should choose *Raise* if $p < 0.8$ and *Not Raise* if it is bigger than 0.8, which in turn means that the

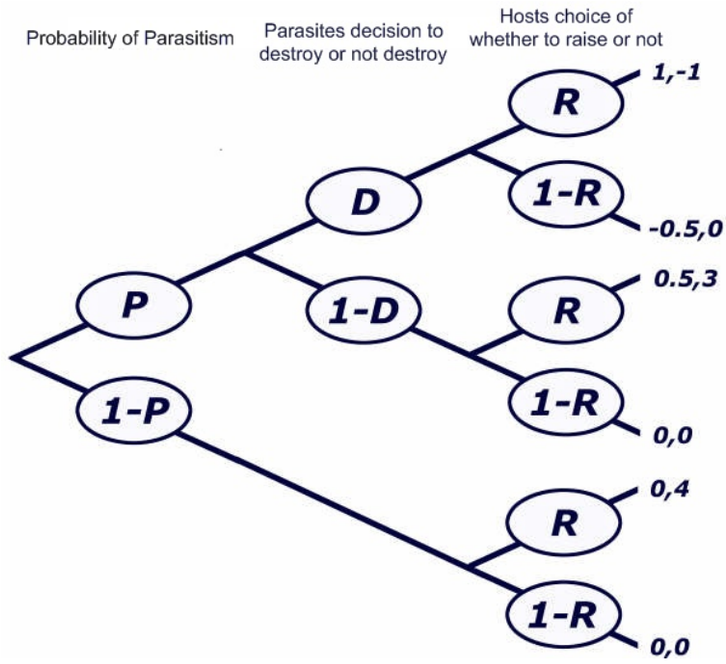


Figure 1.3: Decision Tree of the Second Model with the first decision being the probability of parasitism. The second (if there is a parasite) is the decision whether or not to destroy the host eggs or chick and finally whether the host will raise or not. The numbers at the end of the tree represent the reward to the parasite and reward to the host respectively.

parasite should choose *Not Destroy* if $p > 0.8$ and *Destroy* if it is less than 0.8. So we can see that under different levels of parasitism we can have two different possible outcomes.

1.3.3 Game 3 - Building nature into the model

The problem with the previous model is that we only allow the host to choose raise or not raise. There is no probability of raise if there is an undestroyed nest and don't raise if there is. Assuming the host knows what the parasite has done, we once again get the only solution as (*Don't Destroy*,*Raise*). However, the host could be 'tricked' into believing the destruction occurred naturally. We set up this as in Figure 1.4

We look at a different aspect in this game. We look at the incomplete information question again. In this we give an alternative to the source of the destruction of the host eggs. In which the hosts are destroyed naturally down to one egg (as the parasite would do). If this happens we assume the parasite is destroyed and only one host is left. We say that the host will raise with probability r_f if there is a full nest and r_1 if there is only one (whether this is the parasite or host egg is unknown to the adult host). The parasite as before chooses to destroy with probability d.

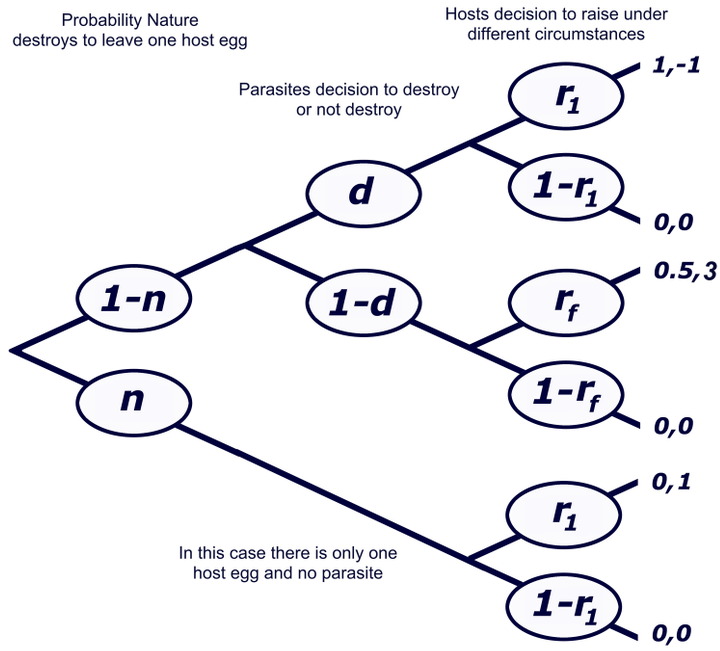


Figure 1.4: Decision Tree of the Third Model with the first decision being the probability of natural destruction. The second (if there is a parasite) is the decision whether or not to destroy the host eggs or chick and finally whether the host will raise or not. The numbers at the end of the tree represent the reward to the parasite and reward to the host respectively.

In this case r_f is always 1 because it will give a better outcome than any $0 \leq r_f < 1$. If the parasite selects $d = 0$ again there is only one way that there can be one egg which is naturally, this means that $r_1 = 1 \iff d = 0$. If the parasite chooses $d = 1$ there are two possibilities that result in having one egg left (where the host will need to choose a value for r_1). The overall reward is

$$n - (1 - n) > 0.$$

So again the host will raise if $n > 0.5$. Therefore if $n > 0.5$ and the parasite chooses $d = 1$ it will get a score of 1. If it chooses $d = 0$ it will get a score of 0.5. If $n \leq 0.5$ the parasite will get -1 if it chooses $d = 1$ and $1/2$ if it chooses $d = 0$. So we get the situation where we have $d = 0, r_1 = 1$ if $n \leq 0.5$ and $d = 1, r_1 = 1$ if $n > 0.5$. This only considers the interaction between parasite chick and the host. These are of course simplistic views of the interactions and the numbers are chosen to show the possibility of interesting results. But as you will see in later chapters they do not go too far from the true values. There are more factors in this game which stem from the interaction of the adults. It may also be prudent to combine the last two games having the probability of parasitism and the probability of natural destruction as factors, which we shall do in the next section.

Chapter 2

Single Interaction Many-Few Model

2.1 The Many-Few Model

2.1.1 Introduction

We now wish to look at this game as a whole, with all the interactions of host and parasite. We shall take some of the elements from the Broom et al model [4] the Servedio-Hauber model [47] and some elements of the previous introductory section on extensive form games. For simplicity we will assume the parasite chicks and adults are the same player, meaning they are essentially players in the same team and know the decisions each other makes. The basic interaction is in Figure 2.1.

We set the model up to allow the host to lay many (*Many*) eggs or just a few (*Few*). The parasite adult then lays one of its own. There is then a period of natural destruction as in Game 3 explained in the previous chapter. The fourth stage allows the parasite chick to destroy or leave the nest alone, with the final stage being the decision by the host to raise or not. We can also set up our costs and rewards as functions of the number of eggs to be raised, destroyed etc as in Table 2.1. In our examples the values of x and y are not numeric. They take the form of the many, few and one/zero decisions. These will be denoted by $M = Many$, $F = Few$, 1 and 0. For example

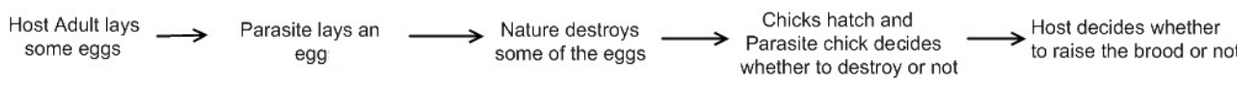


Figure 2.1: Diagram showing the chronological interactions between host and parasite

Table 2.1: Table of Costs and Rewards for both Host and Parasite

Variable	Description
$R_H(x)$	Reward for the host at the end for successfully raising x chicks
$C_L(x)$	Cost to the host for laying x eggs
$C_R(x)$	Cost to the host for raising x chicks
$R_P(x)$	Reward at the end for having x host chicks in the nest with our parasite
$C_D(x, y)$	Cost to the parasite for destroying $X - Y$ eggs when there are X eggs in the nest.

$R_H(M)$ is the reward to the host for raising many eggs and $C_D(F, 0)$ is the cost to the parasite of destroying Few host eggs down to 0 leaving just itself.

We must now set up this game mathematically. First of all we allow the host to make a choice to lay either Many eggs or Few. This decision whether to lay many or few is ultimately the decision we are trying to get out of this game. Many parasites do not show a significant level of destruction as adults as most destroy one host egg to keep the brood size the same (in this model $Many + 1$ is still $Many$ and $Few + 1$ is still Few so this effect of the parasite adult is not significant). We set the next stage of the game up as the natural destruction stage we had in the previous section where if we have many eggs they can be destroyed down to a few or just one egg or the nest is not destroyed(These are to align with the possible choices of the parasite). If the host has chosen to lay just a few eggs there is a possibility again that these are destroyed down to just one or is left not destroyed.

Then the brood hatches and the parasite chick decides whether it is going to destroy or not. If there are a few host eggs it can only choose to destroy all or leave the nest alone. If there are $Many$ host eggs it can decide to leave the nest alone or choose to destroy. If it then chooses to destroy we account for the possibilities that it can destroy all the hosts to leave just itself or just destroy some hosts to leave a few. The final decision the host has is deciding whether it will raise the brood. This will be a different decision whether there are Many, Few or just one egg left in the nest. We use the variables as shown in Table 2.2.

Thus we have two different decisions by the host. One in the first place whether to lay many or few eggs and one in the final stage whether to raise the brood of chicks. Since the host knows how many it has laid in the first place and the parasite is just acting on what happens after the natural destruction stage we can split this into two distinct problems; the Few problem (where the host has chosen to lay few) and the Many problem (where the host has chosen to lay many).

Table 2.2: Table of probabilities, host decisions and parasite decisions

Variable	Description
p_m^m	Probability that natural destruction has not destroyed the eggs in the nest
p_f^m	Probability that natural destruction has taken the many eggs down to just a few
p_1^m	Probability that natural destruction has taken the many eggs down to just one
p_f^f	Probability that natural destruction has not destroyed the eggs in the nest (when their are few eggs in the nest)
p_1^f	Probability that natural destruction has taken the few eggs down to just one
δ_m^m	Probability the parasite will choose not to destroy any host eggs/chicks
δ_f^m	Probability the parasite will choose to destroy the many host eggs/chicks to leave just a few
δ_1^m	Probability the parasite will choose to destroy the many host eggs/chicks to leave itself
δ_f^f	Probability the parasite will choose not to destroy host eggs/chicks (when there are few)
δ_1^f	Probability the parasite will choose to destroy all host eggs/chicks to leave itself
r_m	Probability the host will raise a brood of many eggs
r_f	Probability the host will raise when the brood has few eggs
r_1	Probability the host will raise a brood of just one chick

2.1.2 The Few Problem

The *Few Problem* is the much more simple of the two, since it breaks down very similarly to the models in the previous sections. For this we can simplify the model so $p_f^f = p$, $p_1^f = 1 - p$, $\delta_1^f = d$ and $\delta_f^f = 1 - d$. We assume in this case that since the host has laid few it will raise few so $r_f = 1$. So we simplify $r_1 = r$. This can be displayed in a tree as shown in figure 2.2

We can solve this game by backwards induction. First of all we can make assumptions based upon r and d and see what the other player will play based on these assumptions. For instance if we set $r = 1$ the parasite's expected reward is

$$p(1-d)R_P(F) + pd(R_P(0) - C_D(F,0)). \tag{2.1}$$

We then see when the coefficient of d is positive. Since we know $p > 0$, from equation (2.1) we get

$$R_P(F) + d(R_P(0) - C_D(F,0) - R_P(F)) > 0. \tag{2.2}$$

We only need to look at the part of equation (2.2) $R_P(0) - C_D(F,0) - R_P(F)$ to see if it is positive, if it is then the parasite will destroy and we set $d = 1$ and if it is negative then $d = 0$. This is an answer we would expect as the host is raising in all given situations. The parasite simply chooses which one (destroying down to 1 or leaving as few) will get him the largest reward.

Now we assume $r = 0$, the expected parasite reward in this case becomes

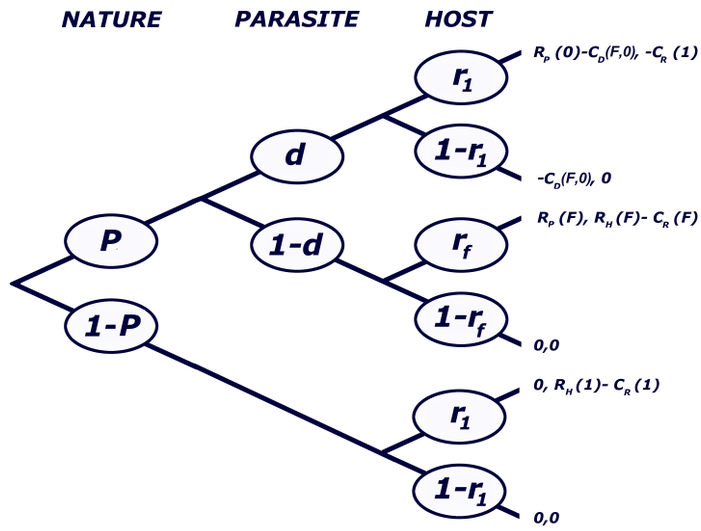


Figure 2.2: Breakdown of the Few Problem - Natural destruction occurs with probability p. If there is no natural destruction the parasite destroys with probability d. The host chooses to raise one egg with probability r_1 whether this is done by natural destruction or the parasite and raises with probability r_f if there is no natural destruction.

$$p(1-d)R_P(F) + pd(-C_D(F,0)). \tag{2.3}$$

Using again the fact that $p > 0$ and rearranging, we get a reward proportional to

$$R_P(F) - d(R_P(F) + C_D(F,0)). \tag{2.4}$$

If we look at equation (2.4) we get that $d = 0$ because any positive d decreases the parasite's outcome. Meaning that in the case if the host never raises the single egg the parasite will never destroy.

We now assume solutions for the parasite and see what we get for the host if we assume $d = 0$. The hosts expected reward is

$$p(R_H(F) - C_R(F)) + (1-p)r(R_H(1) - C_R(1)). \tag{2.5}$$

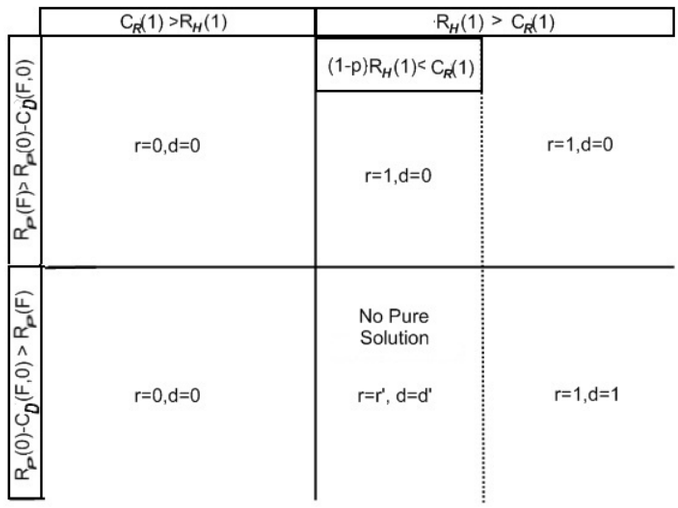
Equation (2.5) implies that $r = 1$ if $R_H(1) - C_R(1) > 0$, and otherwise $r = 0$. The host knows that in this case there is no parasite because we assume it is destroyed in the previous stage. Meaning it will raise as long as it is profitable to do so.

Assuming $d = 1$ gives us equation (2.6) which implies that $r = 1$ if $(1-p)R_H(1) - C_R(1) > 0$,

r	d	Parasite Condition	Host Condition
1	1	$R_P(0) - C_D(F,0) > R_P(F)$	$(1-p)R_H(1) > C_R(1)$
1	0	$R_P(0) - C_D(F,0) < R_P(F)$	$R_H(1) > C_R(1)$
0	0	NA	$R_H(1) < C_R(1)$

Table 2.3: Table showing the conditions under which the host and parasite decisions are made

Figure 2.3: Two dimensional representation of the areas of solution in the Few Problem. The horizontal axis represents the line $R_P(F) = R_P(0) - C_D(F,0)$ with $R_P(F)$ being greater above this line and $R_P(0) - C_D(F,0)$ being greater below this line. The bold vertical axis is the line $C_R(1) = R_H(1)$ and the dotted vertical axis is the line $(1-p)R_H(1) = C_R(1)$



otherwise $r = 0$. The host now is unsure whether there is a parasite in the nest. It will now only raise if the probability of the parasite being there is low enough that it is still profitable to raise the single egg. This gives

$$pr(-C_R(1)) + (1-p)r(R_H(1) - C_R(1)). \tag{2.6}$$

Table 2.3 summarises the above and shows that we can define the two-dimensional area of parameter space shown in Figure 2.3. The horizontal axis represents the line where $R_P(0) - C_D(F,0) = R_P(F)$ and the bold vertical axes being the line $R_H(1) = C_R(1)$. The dotted line is the one where $(1-p)R_H(1) = C_R(1)$. This divides the areas into six distinct regions.

The problem we have is that there is a space without a pure solution. We seek an intermediate value for r and d because we need work out what will happen if $R_P(0) - C_D(F_0) > R_P(F)$ and the host reward falls in the area where $R_H(1) - C_R(1) > 0$ but $(1-p)R_H(1) - C_R(1) < 0$. This is done by setting dummy variables for r and d. We call these r' and d'. Putting $r = r'$ we get,

$$p(1-d)R_P(F) + pd(1-r')(-C_D(F,0)) + pdr'(R_P(0) - C_D(F,0)) = 0 \tag{2.7}$$

$$\implies \quad pR_P(F) + d(-pR_P(F) + p(1-r')(-C_D(F,0)) + pr'(R_P(0) - C_D(F,0))) = 0$$

We need to look at when the coefficient of d in equation (2.7) is 0. This occurs when,

$$r' = \frac{R_P(F) + C_D(F,0)}{R_P(0)}. \tag{2.8}$$

In equation (2.8) when $R_P(F) = R_P(0) - C_D(F,0)$, $r' = 1$ as we need at the boundary. Assuming $d = d'$ gives us

$$pd'r(-C_R(1)) + (1-p)r(R_H(1) - C_R(1)) = 0. \tag{2.9}$$

This means we get our intermediate value when the coefficient of r is zero, giving the following

$$d' = \frac{(1-p)(R_H(1) - C_R(1)}{pC_R(1)} \tag{2.10}$$

which is 0 when $R_H(1) = C_R(1)$ and 1 when $(1-p)R_H(1) = C_R(1)$ which means that it fills the area in figure 2.3 because it has the correct solution on the borders. This means that we have a solution for all possible combinations of host and parasite rewards. This final one is where it is better for the parasite to destroy down to 1. But the host is in an intermediate period where if there was no parasite it would raise one. But because there is a probability of a parasite the decisions are dependent on p and we get the solutions above.

2.1.3 The Many Problem

The many problem is the more complex of the two problems. As we know many eggs have been laid we can simplify some of the terms. We can say that the probability of no natural destruction is now P_m and the probability of destroying down to few and one is P_f and P_1 respectively. We also set up the parasite decisions as just $m, f$and z meaning it destroys leaving many, few and zero hosts respectively. We assume that the host always raises a brood of *Many* chicks ($r_m = 1$). We can display the stages of this game as shown in Figure 2.4.

As with the *Few Problem* we can solve this game via backwards induction. However we must

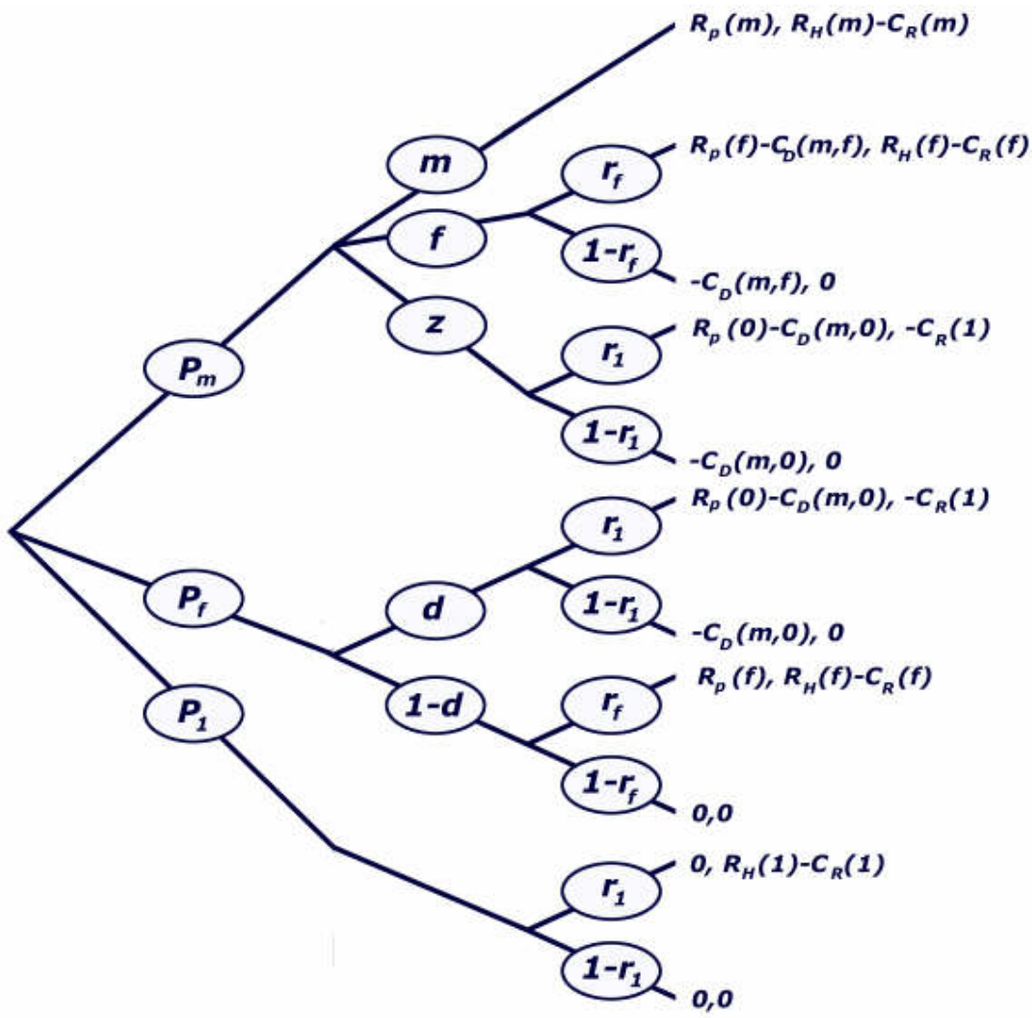


Figure 2.4: Breakdown of the stages of the Many Problem. Starting with Natural Destruction where the *Many* eggs can be left undestroyed P_m, be destroyed down to a few P_f or destroyed to one P_1, then the parasite decision dependent on the outcome of the nature stage. Assuming there was no natural destruction it chooses to not destroy and leave *Many* m, destroy to *Few* f or destroy to leave just itself z. The final decision is the host decision to raise or not depending on how many are left.

make combinations of assumptions since there are more branches to the tree. We start by finding areas of solution for the parasite. If we now assume $r_1 = 0$ and $r_f = 0$, we get a parasite reward of

$$P_m(m(R_P(M)) + f(-C_D(M,F)) + z(-C_D(M,0))) + P_f(d(-C_D(F,0))) \tag{2.11}$$

which is maximised if $m = 1, f = 0, z = 0$ and $d = 0$, so that in the case where there are many eggs the parasite chooses m. This is interpreted as the host only ever raising $Many$ eggs. Because of this the parasite outcome will be negative. If we now assume $r_1 = 1$ and $r_f = 0$ (which is not intuitively a sensible answer but we must cover all the bases) we get a parasite reward of

$$P_m(m(R_P(M)) + f(-C_D(M,F)) + z(R_P(0) - C_D(M,0))) + P_f(d(R_P(0) - C_D(F,0))).$$

This means that $d = 1$ and $f = 0$ always and we have a condition on whether the parasite chooses m or z

$$\begin{aligned} R_P(M) &> R_P(0) - C_D(M,0) \implies m = 1, z = 1 \\ R_P(M) &< R_P(0) - C_D(M,0) \implies m = 0, z = 1 \end{aligned} \tag{2.12}$$

This is equivalent to the Few problem as we have essentially removed the choice for the host of raising a few. So the parasite will destroy down to 1 from many if it is profitable to do so.

Assuming that $r_1 = 0$ and $r_f = 1$ the parasite reward becomes

$$P_m(m(R_P(M)) + f(R_P(F) - C_D(M,F)) + z(-C_D(M,0))) + P_f(d(-C_D(F,0))). \tag{2.13}$$

Equation (2.13) is maximised when $d = 0$ and $z = 0$, and we have a condition which tells us whether the parasite chooses m or f given in equation (2.14).

$$\begin{aligned} R_P(M) &> R_P(F) - C_D(M,F) \implies m = 1, f = 0, \\ R_P(M) &< R_P(F) - C_D(M,F) \implies m = 0, f = 1. \end{aligned} \tag{2.14}$$

Which is equivalent to the Few problem replacing the destruction down to 1 from Few with the destruction down to Few from $Many$. Meaning the parasite will destroy down to few if it is profitable to do so.

Finally we assume that both r_1 and r_f are equal to 1. The parasite reward in this case becomes

$$P_m(m(R_P(M))+f(R_P(F)-C_D(M,F))+z(R_P(0)-C_D(M,0)))+P_f(d(R_P(0)-C_D(F,0))) \quad (2.15)$$

In this case d is always equal to 1. but we now have three conditions to find the values of m, f and z.

$$R_P(M) > R_P(F) - C_D(M,F)\, and\, R_P(M) > R_P(0) - C_D(M,0) \implies m = 1, f = 0, z = 0,$$

$$R_P(F)-C_D(M,F) > R_P(M) and R_P(F)-C_D(M,F) > R_P(0)-C_D(M,0) \implies m = 0, f = 1, z = 0,$$

$$R_P(0)-C_D(M,0) > R_P(M) and R_P(0)-C_D(M,0) > R_P(F)-C_D(M,F) \implies m = 0, f = 0, z = 1.$$

$$\begin{aligned}
R_P(M) > R_P(F) - C_D(M,F) \quad and \quad R_P(M) > R_P(0) - C_D(M,0) &\implies m = 1, f = 0, z = 0, \\
R_P(F) - C_D(M,F) > R_P(M) \quad and \quad R_P(F) - C_D(M,F) > R_P(0) - C_D(M,0) &\implies m = 0, f = 1, z = 0, \\
R_P(0) - C_D(M,0) > R_P(M) \quad and \quad R_P(0) - C_D(M,0) > R_P(F) - C_D(M,F) &\implies m = 0, f = 0, z = 1.
\end{aligned} \quad (2.16)$$

We must now find the solutions for the host based upon the parasite choices. Assuming $m = 1$ and $d = 0$ gives us an expected reward for the host of

$$P_m(R_H(M) - C_R(M)) + P_f(r_f(R_H(F) - C_R(F))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))) \quad (2.17)$$

Equation (2.17) implies that $r_f = 1$ (since we still assume $R_H(F) > C_R(F)$) and

$$R_H(1) > C_R(1) \implies r_1 = 1,$$

$$R_H(1) < C_R(1) \implies r_1 = 0.$$

Now assume $f = 1$ and $d = 0$, we get the following expected outcome for the host

$$P_m(R_H(F) - C_R(F)) + P_f(r_f(R_H(F) - C_R(F))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))) \quad (2.18)$$

Equation (2.18) implies that $r_f = 1$,

$$R_H(1) > C_R(1) \implies r_1 = 1$$

and

$$R_H(1) < C_R(1) \implies r_1 = 0.$$

We now assume $z = 1$ and $d = 0$ this gives us the expected outcome for the host as

$$P_m(R_H(F) - C_R(F)) + P_f(r_f(R_H(F) - C_R(F))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))) \quad (2.19)$$

which again means $r_f = 1$ but we get a different condition on r_1 where

$$(1 - (P_m + P_f))R_H(1) > (1 - P_f))C_R(1) \implies r_1 = 1,$$

and

$$(1 - (P_m + P_f))R_H(1) < (1 - P_f))C_R(1) \implies r_1 = 0.$$

Now we must make similar calculations with $d = 1$. First of all assume $m = 1$ and $d = 1$ where we get the host's expected reward as

$$P_m(R_H(M) - C_R(M)) + P_f(r_1(-C_R(1))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))). \quad (2.20)$$

Since we assume the host destroys to none when there are few and leaves as many when there are many in the nest, there is no way to get to few eggs so there is no condition on r_f here. However we get the following conditions on r_1 from equation (2.20). In which

$$(1 - (P_m + P_f))R_H(1) > (1 - P_m))C_R(1) \implies r_1 = 1$$

and

$$(1 - (P_m + P_f))R_H(1) < (1 - P_m))C_R(1) \implies r_1 = 0.$$

Assuming $f = 1$ and $d = 1$ the host reward becomes

$$P_m(R_H(F) - C_R(F)) + P_f(r_1(-C_R(1))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))). \quad (2.21)$$

In this case r_f is always equal to 1,

$$1 - (P_m + P_f))R_H(1) > (1 - P_m))C_R(1) \implies r_1 = 1$$

and

$$(1 - (P_m + P_f))R_H(1) < (1 - P_m))C_R(1) \implies r_1 = 0.$$

Finally assuming $z = 1$ and $d = 1$, the host's expected reward becomes

$$P_m(\rho - C_R(1)) + P_f(r_1(-C_R(1))) + (1 - (P_m + P_f))(r_1(R_H(1) - C_R(1))). \tag{2.22}$$

As before we have no condition on r_f but we have

$$(1 - (P_m + P_f))R_H(1) > C_R(1) \implies r_1 = 1$$

and

$$(1 - (P_m + P_f))R_H(1) < C_R(1) \implies r_1 = 0.$$

r_1	r_f	d	mfz	$R_H(1)\ v\ C_R(1)$	$R_P(M)\ v$ $R_P(F) - C_D(M,F)$	$R_P(M)\ v\ R_P(F) - C_D(M,F)$ $v\ R_P(0) - C_D(M,0)$	$R_P(M)\ v$ $R_P(0) - C_D(M,0)$	$R_P(F)$ $v\ R_P(0) - C_D(F,0)$
0	1	0	m	$R_H(1) < C_R(1)$	$>$	NA	NA	NA
1	1	0	m	$R_H(1) > C_R(1)$	NA	$R_P(M)$	NA	$>$
0	1	0	f	$R_H(1) < C_R(1)$	$<$	NA	NA	NA
1	1	0	f	$R_H(1) > C_R(1)$	NA	$R_P(F) - C_D(M,F)$	NA	$>$
1	1	0	z	$(1-(P_m+P_f)R_H(1) > (1-P_f)C_R(1)$	NA	$R_P(0) - C_D(M,0)$	NA	$>$
1	1	1	m	$(1-(P_m+P_f)R_H(1) > (1-P_m)C_R(1)$	NA	$R_P(M)$	NA	$<$
1	1	1	f	$(1-(P_m+P_f)R_H(1) > (1-P_m)C_R(1)$	NA	$R_P(F) - C_D(M,F)$	NA	$<$
1	1	1	z	$(1-(P_m+P_f)R_H(1) > C_R(1)$	NA	NA	NA	$<$
1	0	1	m	$(1-(P_m+P_f)R_H(1) > (1-P_m)C_R(1)$	NA	NA	$>$	NA
1	0	1	f	$(1-(P_m+P_f)R_H(1) > (1-P_m)C_R(1)$	NA	NA	$<$	NA
1	0	1	z	$(1-(P_m+P_f)R_H(1) > (1-P_f)C_R(1)$	NA	$R_P(0) - C_D(M,0)$	NA	NA

Table 2.4: Table showing areas where solutions can be found for r_1,r_f,d and the case where the parasite chooses whether to destroy down to many m, few f or no z host eggs.

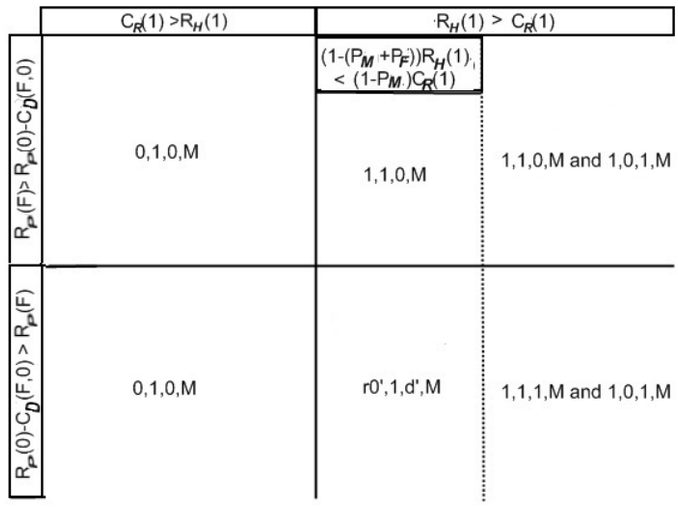


Figure 2.5: Two dimensional representation of the areas of solution in the Many Problem when $R_p(M) > R_p(F) - C_D(M,F), R_p(M) > R_p(0) - C_D(M,0)$. The horizontal axis represents the line $R_P(F) = R_P(0) - C_D(F,0)$ with $R_P(F)$ being greater above this line and $R_P(0) - C_D(F,0)$ being greater below this line. The bold vertical axis is the line $C_R(1) = R_H(1)$ and the dotted vertical axis is the line $(1-(P_m+P_f))R_H(1) = (1-P_m)C_R(1)$

Table 2.4 gives us three possible situations to look at. One aspect will be concerned with the host reward, the other two are to do with the parasite reward. One of these looks at the decision made when there is natural destruction to few and the final aspect looks at the occasion when there is no natural destruction. This is split into three distinct sections dependent on whether $R_p(M)$, $R_p(F) - C_D(M,F)$ or $R_p(1) - C_D(M,0)$ is the largest. For this reason we look at these individually. First of all when $R_p(M)$ is the largest.

We get areas of solution as with the *few problem* as shown in Figure 2.5, where for example $110M$ means that $r_1 = 1$, $r_f = 1$, $d = 0$ and the choice of m, f or z being m. This corresponds to one of the rows in Table 2.4. Like the Few Problem we have *empty* regions we must fill; i.e. do this by considering the intermediate values which we shall call r_1' and d'.

Let $d = d'$, $r_f = 1$, $m = 1$, the expected host reward then becomes

$$P_f((1-d')(R_H(F)-C_R(F))+d'r_1(-C_R(1)))+P_m(R_H(M)-C_R(M))+(1-(P_m+P_f))r_1(R_H(1)-C_R(1))). \tag{2.23}$$

The coefficient of r_1 in equation (2.23) is zero when

$$d' = \frac{(1-(P_m+P_f))(R_H(1)-C_R(1))}{P_f C_R(1)}.$$

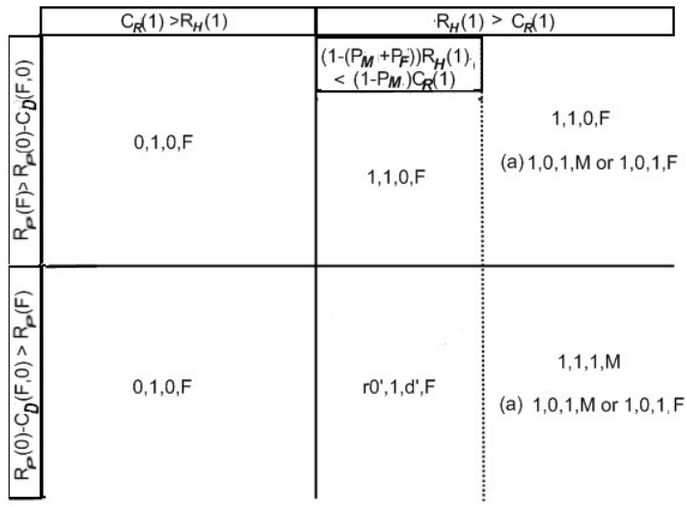


Figure 2.6: Two dimensional representation of the areas of solution in the Many Problem when $R_p(F) - C_D(M,F) > R_p(M), R_p(F) - C_D(M,F) > R_p(1) - C_D(M,0)$. The horizontal axis represents the line $R_P(F) = R_P(0) - C_D(F,0)$ with $R_P(F)$ being greater above this line and $R_P(0) - C_D(F,0)$ being greater below this line. The bold vertical axis is the line $C_R(1) = R_H(1)$ and the dotted vertical axis is the line $(1-p_f)R_H(1) = C_R(1)$. (a) Where $101M$ is a solution if $R_p(M) > R_p(0) - C_D(M,0)$ and $101F$ is a solution if $R_p(M) < R_p(0) - C_D(M,0)$

When $d' = 0$ gives $R_H(1) = C_R(1)$ and when $d' = 1$ this means $(1-(P_m+P_f))R_H(1) = (1-P_m)C_R(1)$ which covers the area required.

Now assuming $r_1 = r_1'$, $r_f = 1$, $m = 1$, the parasite reward is

$$P_m(R_P(M)) + P_f\left[d\left((1-r_1')(-C_D(F,0)) + r_1'(R_P(1) - C_D(F,0))\right) - (1-d)R_P(F)\right], \qquad (2.24)$$

which like the few problem gives us

$$r_1' = \frac{R_P(F) + C_D(F,0)}{R_P(1)},$$

which is 1 when $R_P(F) = R_P(1) - C_D(F,0)$ so this will fill the gap as required.

The area where $R_P(F) - C_D(M,F)$ is the largest of the three rewards gives us the areas as shown in Figure 2.6. Where $101M$ and $101F$ are all possible solutions in the top right area.

We again need to find r_1' and d' shown in the bottom middle area. Letting $d = d', r_f = 1, f = 1$, the expected reward is

$$P_f((1-d')(R_H(F)-C_R(F))+d'r_1(-C_R(1)))+P_m(R_H(F)-C_R(F))+(1-(P_m+P_f))r_1(R_H(1)-C_R(1))) \tag{2.25}$$

If we set the coefficient of r_1 equal to 0 in equation (2.25) we get

$$d' = \frac{(1-(P_m+P_f))(R_H(1)-C_R(1))}{P_f C_R(1)}$$

and as before, because this second aspect has no effect on the interaction when there are *few* eggs. This runs from $R_H(1) = C_R(1)$ (when $d = 0$) to $(1-(P_m+P_f))R_H(1) = (1-P_m)C_R(1)$.

Now assume $r_1 = r_1'$, $r_f = 1$, $m = 1$ the host reward becomes

$$P_m(R_P(F) - C_D(M,F)) + P_f\left[d\left(1-r_1'\right)(-C_D(F,0)) + r_1'(R_P(1) - C_D(F,0))\right) + (1-d)R_P(F)\right] \tag{2.26}$$

which implies that

$$r_1' = \frac{R_P(F) + C_D(F,0)}{R_P(1)},$$

which is 1 when $R_P(F) = R_P(1) - C_D(F,0)$ and this fills the gap as required.

Finally we look at the area where $R_P(1) - C_D(M,0)$ is the largest. This means that we can cover the areas shown in Figure 2.7.

Again we need to find r_1' and d'. Letting $d = d'$, $r_f = 1$, $z = 1$ the expected host reward is

$$P_f((1-d')(R_H(F)-C_R(F))+d'r_1(-C_R(1)))+P_m r_1(-C_R(1))+(1-(P_m+P_f))r_1(R_H(1)-C_R(1))). \tag{2.27}$$

Similarly to before from equation (2.27) we get

$$d' = \frac{(1-(P_m+P_f))R_H(1) - (1-P_f)C_R(1)}{P_f C_R(1)},$$

which when $d' = 1$ is $(1-(P_m+P_f))R_H(1) = C_R(1)$ and when $d' = 0$ is $(1-(P_m+P_f))R_H(1) = (1-P_f)C_R(1)$.

Now assume $r_1 = r_1'$, $r_f = 1$, $z = 1$. The parasite reward in this case is

$$P_m(R_P(0)-C_D(M,0))+P_f(d(1-r_1'(-C_D(F,0))+r_1'(R_P(0)-C_D(F,0)))-(1-d)R_P(F)), \tag{2.28}$$

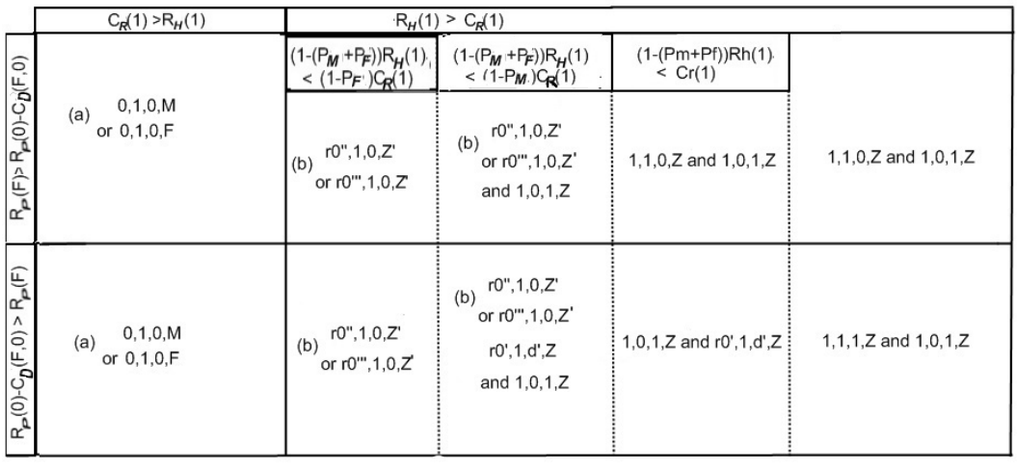


Figure 2.7: Two dimensional representation of the areas of solution in the Many Problem when $R_P(0) - C_D(M,0) > R_P(M), R_P(0) - C_D(M,0) > R_P(F) - C_D(M,F)$. The horizontal axis represents the line $R_P(F) = R_P(0) - C_D(F,0)$ with $R_P(F)$ being greater above this line and $R_P(0) - C_D(F,0)$ being greater below this line. The bold vertical axis is the line $C_R(1) = R_H(1)$ and the dotted vertical axes are the lines $(1-p_f)R_H(1) = C_R(1)$, $(1-p_m)R_H(1) = C_R(1)$ and $(1-p_m+p_f)R_H(1) = C_R(1)$. (a) Where $010M$ is a solution if $R_P(M) > R_P(F) - C_D(M,F)$ and $010F$ is a solution if $R_P(M) < R_P(F) - C_D(M,F)$. (b) Where $r_1'', 1, 0, Z'$ is a solution if $R_P(M) > R_P(F) - C_D(M,F)$ and $r_1''', 1, 0, Z'$ is a solution if $R_P(M) < R_P(F) - C_D(M,F)$.

which gives

$$r_1' = \frac{R_P(F) - C_D(F,0)}{R_P(1)}$$

which is 1 when $R_P(F) = R_P(1) - C_D(F,0)$.

We now need to find r_1'', r_1''' and z',

Let $d = 0, r_f = 1, z = z'$, we only need concentrate on the parts of the host reward which contain r_1; in this case we get

$$P_m(z' r_1(-C_R(1))) + ((1-(P_m+P_f))r_1(R_H(1) - C_R(1))), \tag{2.29}$$

which gives

$$z' = \frac{(1-(P_m+P_f))(R_H(1) - C_R(1))}{P_m C_R(1)}, \tag{2.30}$$

where the value is $m' = 1 - z'$ or $f' = 1 - z'$ depending on the equality which is larger $R_P(M)$ or $R_P(F) - C_D(M,F)$, where if $R_P(M) > R_P(F) - C_D(M,F)$ then $m' = 1 - z'$, $f' = 0$ and vice-versa.

Thus there are two values for r_1 for the two different cases which we will label r_1'' for the m'

case and r_1''' for the f' case.

To find r_1'' let $d = 0$, $r_f = 1$ and $r_1 = r_1''$. The parasite reward becomes

$$P_m((1-z)R_P(M) + z((1-r_1'')(-C_D(M,0))) + r_1''(R_P(0) - C_D(M,0))) = 0. \tag{2.31}$$

From equation (2.31) we get

$$r_1'' = \frac{R_P(M) + C_D(M,0)}{R_P(1)}. \tag{2.32}$$

For r_1''' we will have the same equation as (2.32) but replacing $R_P(M)$ with $R_P(F) - C_D(M,F)$, i.e.

$$r_1''' = \frac{(R_P(F) - C_D(M,F)) + C_D(M,0)}{R_P(1)}. \tag{2.33}$$

2.2 The overall problem

We have found solutions for both the Many Problem and the Few Problem. If we assume the expected outcome for the host is H_m from the *Many problem* and the expected reward for host is H_f for the few problem then we get the following.

$$H_m - C_L(m) > H_f - C_L(f) \implies Many$$

and

$$H_f - C_L(f) > H_m - C_L(m) \implies Few.$$

If there is equality the choice made does not matter.

2.3 Discussion

We now have a comprehensive reward scheme for the host under certain conditions. We can work out whether choosing to lay many eggs or few eggs is the best course of action. This is a good base model as it shows we can achieve strong solutions under relatively simple conditions. However many of the natural elements of the interaction cannot feature due to our assumptions.

For instance, our $Many + 1 = Many$ principle makes it very easy for the parasite to disguise

itself without having to do any destruction as an adult so the possibility of abandoning the nest early is not included. Also, the fact that $Many - 1 = Many$ means the egg ejection principle (which is one of the main defences some hosts have for dealing with parasitical behaviour [38], [28], [34],[35]) cannot be implemented easily and to full effect.

Because of this we must move on to allowing the host to lay a number of eggs. This will then hopefully force the parasite adult into some destruction in order to disguise and allow the host to eject any eggs it deems to be a parasite.

These solutions give fairly expected rewards in that destruction leads to hosts not wishing to lay unless the probability of natural destruction is high enough. If the probability of natural destruction is high enough then it may raise the brood because it has a higher chance of raising some of its own chicks. The greatest reward for the parasite occurs when the probability of natural destruction is high. If the parasite is lucky and there is no natural destruction in this case it can destroy the nest itself and reap the benefits.

Some of the mixed solutions occur because of the natural destruction parameters. There will be an area when there is no distinct choice between raise and not raise or destroy and not destroy. This choice will be dependent on the natural destruction parameters and the reward to the opponent.

We have assumed that the hosts only have the capacity for differentiating between many, few and one egg. This is not necessarily the case. The model gives us a good indication of the power of using extensive form games to model this interaction. However we need to produce a more realistic model. For this reason we must allow the host to raise any number of eggs. This also takes away the $Many + 1 = Many$ issue as well as allowing us to account for the other aspects of the interaction seen in nature.

Chapter 3

The Single Interaction Model

The Many-Few model gave us the initial break down of the game but it did not give us the freedom to add in the possibilities of egg-ejection and other decisions. We can now allow the host and parasite to make decisions based upon the number of eggs. We allow the host to lay some number n of eggs and base the rest of the decisions upon this. We define the interaction in Figure 3.1. The first stage is a decision by the host bird at the beginning of the game to lay a certain number of eggs. After this occurs, there is a period of time in which a single parasite can visit the nest. If it does, then it decides whether to lay an egg. If it does lay an egg it also has the option to eject some of the host eggs from the nest. If it does not lay an egg then the host may continue just as if the parasite had not been there. Following this the host can make one of three decisions; it can abandon the nest, eject an egg in an attempt to remove the parasite and continue to nurture the nest, or just continue to nurture the nest. This then goes on to the hatching stage; once the eggs hatch the parasite chick makes another decision whether to destroy/eject/bury any number of the host chicks or unhatched eggs. The final decision is that of the host whether to raise the brood depending on the number of chicks in the nest and the likelihood that it is raising a parasite.

The problem we must solve when looking at this model is the fact that at any stage the host does not know where it is on the game tree. For example, if there are four eggs in the nest in the middle of the game, are they all host eggs, or is one of them a parasite? The host will make its decision based upon the probability that there is a parasite given the number of eggs observed. Thus the standard backward induction methods will not work as information sets contain more than a single point on the tree, and we have a game of imperfect information where not only is the position on the tree uncertain, but the probability of being in certain positions depends upon

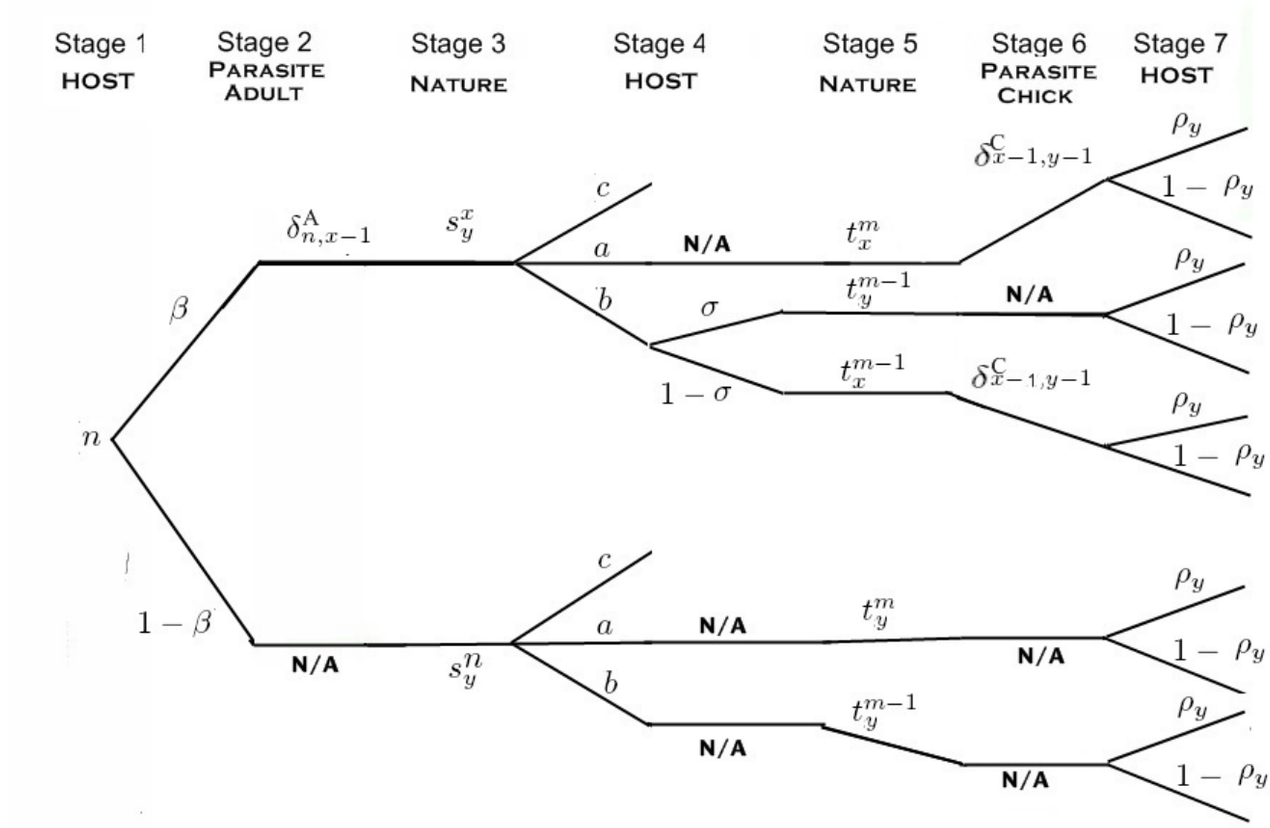


Figure 3.1: Diagram of Decisions by Host and Parasite. The decision is the host laying n eggs. The next shows the probability of parasitism β with a two way split. If there is a parasite adult it chooses to destroy the n down to some $x-1$. From this number Nature destroys the eggs which have made it down to some y eggs. We have a stage where the host can make one of three choices (in the order shown in the diagram these are c - Abandon, a-Do nothing and b attempt ejection). We say the number left from this is m (which could be y or $y-1$. We have another stage of natural destruction down to some x eggs. The parasite chick will destroy down to $y-1$

earlier decisions. There is thus an interaction between earlier and later decisions, with the optimal choice in each depending on that in the other. This is displayed graphically in figure 3.1.

S1 This is the first and overall main decision by the host, which is the choice of how many eggs (n) to lay at the beginning of the cycle, which can theoretically be anything from one upwards. Of course in practice there will be a certain maximum number the host will be able to lay, but at this stage we shall allow for any number, and the host will be prevented from laying large numbers by increasingly prohibitive costs. At this point the host will incur a laying cost which we shall call $C_L(n)$. Biologically speaking this cost represents the use of resources in laying eggs in the current nest at a cost to other activities which may affect survival or the ability to lay more eggs at a later date when perhaps the situation is better for the host.

S2 The value β is the probability that a parasite visits the nest and lays an egg. In the case where there is no parasite we skip to S3. Stage 2 is the decision as to whether the adult parasite will destroy some, all or none of the host eggs and lay one of its own. This decision by the parasite will be denoted as $\delta^A_{n,x-1}$, where a value for $\delta^A_{n,x-1}$ will be given for all $x \in [1, n+1]$. These values will signify the probability that if the adult parasite sees n eggs it will destroy $n-x-1$ to leave x (including its own), therefore $\sum_{x=1}^{n+1} \delta^A_{n,x-1} = 1$. It will usually be the case that for one value of x, $\delta^A_{n,x-1} = 1$ and for the rest this will be zero. If it does destroy down to a total of x eggs it will incur the cost $C_{DA}(n-x-1)$, the cost of destruction for the adult parasite. This relates to the fact that the parasite must make an effort in order to destroy some of the host eggs; this could relate to a loss of energy or time. The loss of time could be important as this may lead to the parasite being discovered by the host. Similarly, the more the nest is disturbed, the greater the chance of alerting the host.

S3 This is the first of two natural destruction stages, and it affects both host eggs and the one parasite egg (if there is such an egg). If there are x host eggs in the nest and no parasite then the probability that y host eggs survive is given as s^x_y. If the nest has $x-1$ host eggs plus a parasite, we set the probability that y of those eggs are left after S3 again as s^x_y. If y eggs are left in total at this point then we assume that the parasite has a probability of survival of $\frac{y}{x}$ (i.e. the parasite has the same chance of survival as each host egg). This means that the overall probability of survival for the parasite is $\sum_{y=0}^{x} s^x_y \frac{y}{x}$.

Natural destruction could occur due to nest predation, bad weather or poor parental care.

If it is predation, usually the whole nest will be lost, and an alternative idea would be to simplify our model by allowing only no or full destruction. However, we want to maintain the flexibility of a more general model.

S4 This is a decision that the host makes before hatching occurs. This occurs a while after laying when some natural destruction may have occurred and is in the time-period after which any parasite must have arrived (a later parasite's egg would not hatch, because host incubation is too far advanced).

The host makes one of three decisions:

(a) Leave the nest alone, so choosing $a = 1$. This means that the host will do nothing and leave the nest as it is.

(b) Eject one egg ($b = 1$). If the host believes there may be a parasite then it can eject one egg, which will be the correct egg (the parasite) with probability σ, if there is indeed a parasite.

(c) Abandon the nest ($c = 1$).

Choosing one of these decisions implies that all of the others are 0. i.e. if $a = 1$ this implies $b = c = 0$. We label the number of eggs remaining at the end of this stage m.

S5 This is the second natural destruction stage and has the same basis as S3, however we label the probability of destruction as t_x^m.

S6 This is a decision by the parasite chick to destroy a number of the eggs or chicks. We use the term $\delta^C_{x-1,y-1}$ to define the decision to destroy $x - y$ eggs (i.e. $\delta^C_{x-1,y-1} = 1$ iff $x - y$ are destroyed, and otherwise $\delta^C_{x-1,y-1} = 0$), so leaving $y - 1$ host eggs (so y eggs in total) in the nest if there are $x - 1$ host eggs in the nest at this stage. If it does this, then as before it will incur the cost $C_{DC}(x - y)$. This cost could be described as before both in terms of the amount of energy exerted to destroy or eject an egg, or the time in which it takes to eject an egg. The time factor may be important because it may result in detection by the adult which we would then assume may kill the parasite chick or abandon the nest.

S7 This is the final decision of the host whether to raise the full brood or not. If the number of eggs that have made it to this stage is y, then it will incur a cost of $C_R(y)$ if it chooses

to raise. The parasite will receive a reward depending on how many host eggs there are in the nest. This is denoted as $R_P(y-1)$. The host will receive a reward ($R_H(z)$) depending on how many z of its own eggs make it to this stage. The host's decision will be denoted by ρ_y, the probability that given there are y chicks in the nest at this final stage, the host will raise them. In most cases this probability will either be one or zero. Where $\rho_y = 1$ it means that the host will always raise if there are y chicks in the nest and where $\rho_y = 0$ it means that the host will never raise if there are y chicks in the nest. The fitness cost to raising the parasite may be higher for a host parent than the cost of raising a chick of its own, this extra cost being denoted by ξ, so that the cost of the parasite chick is equivalent to ξ host chicks. Thus if there is a parasite the cost to the host becomes $C_R(y-1+\xi)$. This cost represents the physical exertion the host must put out in order to feed and otherwise raise the brood. Obviously the larger the brood the more food it will have to gather and the harder it will be to get the whole brood raised, and this cost may be in decreased probability of successful raising of the brood, or in its own survival chances.

Note that we allowed egg ejection in Stage 4 but not chick ejection in Stage 7. As shown in Planqué et al [40] chick-rejection is not cost effective due to the relative size of the birds in this case and is also not seen in nature. So for the purposes of simplicity we discard the possibility of ejecting the chick. As in the previous section we will have a number of rewards for the host and parasite as well as decisions the host and parasite will make during the game. These can be seen in Table 3.1.

3.1 Breaking the model down

As we stated earlier this cannot be broken down using the standard backward induction methods directly, however it is possible to solve this problem numerically, by feeding forward information from the start of the game with various possibilities, and finding consistent solutions when feeding back from the end of the game in the standard way . In order to compute this model we break it down into two games. One which runs from S4 to S7 which we shall call the Chick Game and another which runs from S1 to S4 which we shall call the Adult Game. This will mean there is an interaction between the games at S4, where the decision in S4 will be determined by the outcomes and decisions in the stages after this. The decisions made in S1 and S2 will be determined by the

Table 3.1: Table of Costs, Rewards and Decisions for the Host and Parasite in the Single Interaction Game.

Parameter	Description
$R_H(x)$	Reward to the host for having x chicks in the nest at the end of the game
$R_P(x)$	Parasite reward when there are x host chicks with the parasite at the game's end
$C_R(x)$	Cost to the host for raising x chicks in the nest at the end of the game
$C_L(x)$	Cost to the host for laying x eggs in the beginning of the game
C_a	Cost of abandoning the nest in the middle of the game
C_b	Cost of abandoning the nest at the end of the game
C_E	Cost to the host if it chooses to eject an egg
$C_{DA}(x)$	Cost to the parasite adult for destroying x host eggs
$C_{DC}(x)$	Cost to the parasite chick for destroying x host chicks
ξ	The relative demand on resources of a parasite chick to a host chick
β	Probability that a parasite will visit the nest and lay an egg
s_y^x	Probability that if there are x eggs all but y will be destroyed (Adult Game)
t_x^m	Probability that if there are m eggs all but x will be destroyed (Chick Game)
σ	Probability that the host correctly recognizes the parasite if it chooses to eject in S4
n	Decision of the number of eggs to lay in S1
ρ_y	Decision to raise or not if there are y chicks left at the end
a	Decision to leave the nest alone in S4 (i.e. $a = 1 \Rightarrow$ nest is left alone)
b	Decision to eject one egg in S4 (i.e. $b = 1 \Rightarrow$ eject one egg)
c	Decision to abandon the nest in S4
$\delta^A_{n,x-1}$	Decision by the parasite adult to destroy $n - x$ eggs leaving $x - 1$ host eggs
$\delta^C_{m-1,x-1}$	Decision by the parasite chick to destroy $m - x$ chicks leaving $x - 1$ host chicks

expected outcome of the given decision in S4. This is diagrammed later in figure 3.2 in more detail.

The Chick Game

In this game we require the probability that a parasite egg has made it to stage S4; we call this probability α, which we evaluate in the next section. We finally look at the decision made in S7 and in particular the value of $r(x, y)$, the expected reward for raising a clutch containing x chicks given that y eggs made it to the start of the chick game (whether this contains a parasite being unknown to the host) . To do this we break down the value of $r(x, y))$ into four possibilities; firstly where there was no parasite and then when there is a parasite combined with the three possible host decisions given by $a = 1$ ($b = c = 0$), $b = 1$ ($a = c = 0$) and $a = b = 0$ (meaning that $c = 1$ and the decision to abandon was taken). For example $H_a(x, y)$ is the expected reward to a host if it chooses to raise a clutch of size x, conditional on there originally having been a parasite and the host having made the decision to raise at stage 4. This factors in the various possible events between stages 4 and 7 which could have led to the clutch size reaching x (natural as well as parasite induced) to find the probability of there being a parasite present.

The outcome for the host in the chick game given there is no parasite in the nest is

$$H_0(x,y) = at_x^y(R_H(x) - C_R(x)) + bt_x^{y-1}(R_H(x) - C_R(x)) - cC_a. \tag{3.1}$$

The outcome for the host in the chick game given there is a parasite in the nest and the decision at Stage 4 is $a = 1$ is

$$\begin{aligned} H_a(x,y) = &\sum_{z=x}^{y} t_z^y(\frac{z}{y}\delta^C_{z-1,x-1}(R_H(x-1) - C_R(x+\xi-1)). \\ &+ t_x^y\left(1 - \frac{x}{y}\right)(R_H(x) - C_R(x)) \end{aligned} \tag{3.2}$$

The outcome for the host in the chick game given there is a parasite in the nest and the decision at Stage 4 is $b = 1$ is

$$\begin{aligned} H_b(x,y) = &\sigma t_x^{y-1}(R_H(x) - C_R(x)) \\ &+ (1-\sigma)\Bigg(\sum_{z=x}^{y-1} t_z^{y-1}\frac{z}{y-1}\delta^C_{z-1,x-1}(R_H(x-1) - C_R(x+\xi-1)) \\ &+ t_x^{y-1}(1 - \frac{x}{y-1})(R_H(x) - C_R(x))\Bigg). \end{aligned} \tag{3.3}$$

The outcome for the host in the chick game if the decision at Stage 4 is $c = 1$ is

$$H_c(x,y) = -C_a. \tag{3.4}$$

Combining equations (3.1),(3.2),(3.3) and (3.4) we get

$$r(x,y) = (1-\alpha)H_0(x,y) + \alpha(aH_a(x,y) + b(H_b(x,y) - C_e) + (1-(a+b))H_c(x,y)). \tag{3.5}$$

We can also work out the outcome for the parasite in *Stage 6* given the different decisions, where we assume that m eggs have made it to *Stage 5*. We also assume that x eggs have made it to *Stage 6* with the parasite surviving with probability $\frac{x}{n}$. So the outcome for the parasite if it

chooses to destroy $x - y$ eggs to leave y is

$$\Pi_{x,y} = \rho_y R_P(y) - C_{DC}(x - y). \tag{3.6}$$

where $\Pi_{x,y}$ is the reward to a parasite chick given that it survived to *Stage 6* as one of x eggs and chooses to destroy down to a total of y. In general we will use the symbol Π to represent the reward to the parasite. In particular in addition to $\Pi_{x,y}$, we define Π to be the overall reward to the parasite at the start of the game, $\Pi(x)$ as the expected reward for the parasite if x eggs are in the nest at the start of the Chick game and Π_Γ as the expected reward to the parasite chick given that it survives to *Stage 6* and that it plays the strategy vector Γ (prior to the number of surviving eggs being known).

We can then use this in turn to find the optimal decision for the host in Stage 4.

The Adult Game

We have to use backward induction again to evaluate the Adult Game and we need to look at S4 and with this the Chick Game. In particular we need to work out the decision made at S4 by the host. The host will then make the decisions in the later stages based upon the outcomes from the Chick game. This outcome depends upon the value of α. Using conditional probability we can deduce

$$\alpha = P(Parasite|\ x\ eggs) = \frac{P(Parasite\ \&\ x\ eggs)}{P(x\ eggs)}. \tag{3.7}$$

There are different possibilities of how there came to be x eggs at *Stage 3*, given that n host eggs were laid.

Firstly, there was no parasite in the nest at all and all the destruction was natural, occurring with probability

$$\alpha_0 = (1 - \beta)s_x^n. \tag{3.8}$$

Secondly, there was a parasite and the destruction was caused in part by the parasite and in part by nature with the parasite egg not destroyed, occurring with probability

$$\alpha_1 = \beta \sum_{k=x-1}^{n} \delta_{n,k}^A s_x^{k+1} \left(\frac{x}{k+1}\right). \tag{3.9}$$

Thirdly, there was a parasite and the destruction was caused in part by the parasite and in part by nature with the parasite egg destroyed

$$\alpha_2 = \beta \sum_{k=x}^{n} \delta_{n,k}^{A} s_x^{k+1} \left(1 - \frac{x}{k+1}\right). \tag{3.10}$$

From equations (3.8),(3.9) and (3.10) we get

$$\alpha = \frac{\alpha_1}{\alpha_0 + \alpha_1 + \alpha_2}. \tag{3.11}$$

Equation (3.11) will then give us an outcome for S4 onwards and thus we can find the decision made at S4 by the host. From this we can work out the best decision for the parasite at S6 and so on. We get the following outcome for the parasite if it destroys down to y eggs at Stage 2

$$P(n,x) = \sum_{y=0}^{x} s_y^x \frac{y}{x} (a_y \Pi(y-1) + b_y(1-\sigma)\Pi(y-2)), -C_{DA}(n-x-1) \tag{3.12}$$

where $a_y = 1$ means that the decision from the host in S3 is to leave the nest alone and $\Pi(x)$ is as described above. If the host never raises a brood this could result in a negative outcome for the parasite, however this also results in a game where the host will never raise any of its own chicks, which would most likely mean a nest will not be formed in the first place. This scenario is unlikely, therefore, to correspond to any real situation; in particular the parasite will not make a decision which the host will follow by not raising.

Once we know the decision by the parasite we can also work out the decision from the host in Stage 1.

$$H(n) = (1-\beta) \sum_{y=0}^{n} s_y^n \Omega(y) + \beta \sum_{x=0}^{n} \delta_{n,x}^{A} \sum_{y=0}^{x} s_y^x \Omega(y) \tag{3.13}$$

where $\Omega(x)$ is the expected reward to a host in Stage 4 when there are x eggs.

3.1.1 Computing the Model

Real clutch sizes can be large (up to about 20 chicks for some species like the gray partridge [41]) so the set of possible sequences of events can be extremely large. We have written a set of programs using MATLAB version 7 to compute our solutions of the aspects we wish to get out of

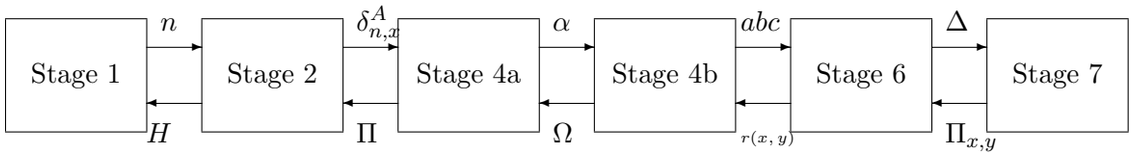


Figure 3.2: Stages of the computer program

this program. These incluse the hosts decisions of n the number of eggs laid and the decision to eject, abandon or not a, b, c and the parasite decisions as to destroy as an adult δ and as a chick γ We created six programs with one feeding information into another, starting from the end of the game first and working backwards. This is illustrated in Figure 3.2 and the code can be found in the Appendix. The arrows in Figure 3.2 pointing left to right represent information being fed into later stages of the program, those from right to left represent the dynamic programming method of finding optimal decisions based upon later ones.

Stage 7

In this part we have all the information necessary to calculate the values of $r(x, y)$ for the host as shown in the previous section. This will also allow us to find the optimal values of ρ_x for each of the possible values of x. All we need to do is compare each $r(x, y)$ to $-C_b$. If it is bigger then we set $\rho_x = 1$, and if it is not then $\rho_x = 0$.

Stage 6

Assume that m eggs have reached *Stage 4* and if i eggs are left after natural destruction then the parasite will choose to destroy leaving γ_i host chicks. We denote Γ as the vector

$$\Gamma = [0, \gamma_1, \gamma_2, \ldots, \gamma_{m-1}]. \tag{3.14}$$

We need to find the best choice of Γ for the parasite (i.e. that which maximises $\Gamma_{x,y}$). The easiest way to do this (mathematically) is to feed all possible values of Γ

$$\begin{array}{cccccc} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ \vdots & & & & \vdots & \\ 0 & 1 & 2 & \dots & m-2 & m-2 \\ 0 & 1 & 2 & \dots & m-2 & m-1 \end{array}$$

into the *Stage 7* program to calculate the expected outcome for every possible decision. Then we select the one which gives the best outcome for the parasite chick.

Note that there is a relationship between the γ's and the δ^C's. The γ's are the actual number of host chicks the parasite will choose to destroy given a number i whereas the δ^C's represent a binary decision. i.e. does the parasite destroy down to x eggs if there are i in the nest. So if $\gamma_i = x$ then this means $\delta^C_{i,x} = 1$ with $\delta^C_{i,y} = 0$ for all $y \neq x$.

Definition 1. *We define* Γ^* *as the value of the vector* Γ *which yields the largest outcome for the parasite chick.*

However since there are $m!$ possible variations of Γ, this poses problems for use on a computer. If $m = 8$ this means we have to run the code 40320 times, which takes approximately 3 minutes using a standard PC. However potentially we need to be able to calculate for much larger values of m, up to about 20 since some hosts will lay this many eggs, and we would have to run the code 2432902008176640000 times. We use an alternative process instead, as follows.

Initially we choose

$$\Gamma = [0, 0, 0, \dots, 0]$$

and calculate the best outcome for place $m-1$

$$\begin{array}{cccccc} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ \vdots & & & & \vdots & \\ 0 & 0 & 0 & \dots & 0 & m-2 \\ 0 & 0 & 0 & \dots & 0 & m-1 \end{array}$$

We select the best of these for the parasite $(0, 0, \dots, \gamma'_{m-1})$ and move to the $m - 2^{th}$ position.

$$\begin{array}{cccccc} 0 & 0 & 0 & \dots & 0 & \gamma'_{m-1} \\ 0 & 0 & 0 & \dots & 1 & \gamma'_{m-1} \\ \vdots & & & & \vdots & \\ 0 & 0 & 0 & \dots & m-3 & \gamma'_{m-1} \\ 0 & 0 & 0 & \dots & m-2 & \gamma'_{m-1} \end{array}$$

We continue down to zero, obtaining

$$\Gamma' = (\gamma'_0, \gamma'_1, \dots, \gamma'_{m-1}).$$

This lowers the amount of γ's we check from $m!$ to $\sum_{x=0}^{m-1} x = \frac{m(m-1)}{2}$ or in the case of $m = 30$ from that very large number to 435.

We proceed to show that (under reasonable conditions) this estimated $\gamma^{'}$ is the same as the *true* γ for our system.

Theorem 1. *If* $R_H(x-1) - C_R(x-1+\xi) < 0$ *then* $\Gamma^* = \Gamma^{'}$*, for all* $1 \leq x \leq m$*.*

The condition $R_H(x-1) - C_R(x-1+\xi) < 0$ for all x means that the parasite has a sufficiently large detrimental effect that the host will always have a negative outcome. Thus if the host was certain that there was a parasite present, abandonment would be the best policy.

Note that Γ^* and $\Gamma^{'}$ are not in general equal because the chick rejection strategy of the parasite chick affects the probability that a nest with a certain number of chicks will actually contain a parasite. For example if the we have a decision in our new $\Gamma^{'}$ which states it is better to destroy down to $y+1$eggs when we have x in nest. We then look at the best decision for $x-1$ which could also in this case be $y+1$. However if we look at both choices at the same time we could have y being the optimal solution (That which is given in Γ^*). This theorem states that under given conditions this situation cannot happen.

Accordingly, any elements of the parasite strategy set can affect a decision of the host against any number of chicks.

Proof. The proof is by induction.

1. First of all we prove that if $[0, \gamma_1, \gamma_2, \dots, \gamma_{m-1}]$ is the true solution that the first cycle will produce $[0, 0, 0, \dots, 0, \gamma_{m-1}]$ in the *quick* solution. i.e. $\gamma'_{m-1} = \gamma_{m-1}$.

If we set $\Gamma^* = [0, \gamma_1, \gamma_2, \ldots, \gamma_{m-1}]$ (the true solution) and $\Gamma' = [0, 0, 0, \ldots, 0, \gamma'_{m-1}]$, we get the following outcomes for Π_γ and $\Pi_{\gamma'}$.

$$\begin{aligned}
\Pi_{\Gamma^*} &= \sum_{x=1}^{m} t_x^m \sum_{y=1}^{x-1} \delta^C_{x-1,y-1}(\rho_y R_p(y-1) - C_{DC}(x-y)) \qquad (3.15)\\
&= t_m^m(\rho_{\gamma_{m-1}+1} R_p(\gamma_{m-1}) - C_{DC}(m-\gamma_{m-1}))\\
&\quad + t_{m-1}^m(\rho_{\gamma_{m-2}+1} R_p(\gamma_{m-2}) - C_{DC}(m-1-\gamma_{m-2}))\\
&\quad + \cdots + t_1^m(\rho_{\gamma_1} R_p(0) - C_{DC}(0))
\end{aligned}$$

$$\begin{aligned}
\Pi_{\Gamma'} &= \sum_{x=1}^{m} t_x^m \sum_{y=1}^{x-1} \delta^{C'}_{x-1,y-1}(\rho_y R_p(y-1) - C_{DC}(x-y)) \qquad (3.16)\\
&= t_m^m(\rho_{\gamma'_{m-1}+1} R_p(\gamma'_{m-1}) - C_{DC}(m-\gamma'_{m-1}))\\
&\quad + \sum_{x=0}^{m} t_x^m(\rho_1 R_p(0) - C_{DC}(x-1))
\end{aligned}$$

where the δ^C's come from the γ's in Γ^* and the $\delta^{C'}$'s come from the γ''s in Γ' as previously described.

The only place where both γ_{m-1} and γ'_{m-1} appear is in the first term of each expression. So the best choice of γ'_{m-1} will be the same as the true value as long as $\rho_{\gamma_{m-1}+1} = \rho_{\gamma'_{m-1}+1}$.

So we must look at the host outcome in Stage 7. Without loss of generality we assume $a = 1$ (an almost identical argument works for $b = 1$). We also need only to look at the parts where the decision of the parasite affects the decision in this final stage. Note if $\rho_x = 0\ \forall x \in (0, y)$ then it is clear that $\gamma_y = \gamma'_y = y$ since the host will never raise.

Here the outcome for the host is $r(x) = H_0(x) + \alpha H_a(x)$ where equation (3.1) becomes

$$H_0(x) = (1-\alpha)\left(a t_x^m (R_H(x) - C_R(x)\right). \qquad (3.17)$$

Which is not affected by the parasite and equation (3.2) is calculated as

$$H_a(x) = \sum_{z=x}^{m} t_z^m \frac{z}{m} \delta^C_{z-1,x-1}(R_H(x-1) - C_R(x+\xi-1)) + t_x^m(1-\frac{x}{m})(R_H(x) - C_R(x)) \tag{3.18}$$

where only part of $r(x)$ affected by the parasite is $\sum_{z=x-1}^{m} t_z^m \frac{z}{m} \delta^C_{z-1,x-1}(R_H(x-1) - C_R(x+\xi-1))$ and we shall denote $r(x)$ minus this expression by r_{NP}. In addition we shall also assume $r_{NP} > 0$, since otherwise unparasitised nests would not be profitable.

With the given values from Γ^*, the above formula rearranges to

$$t_m^m(R_H(\gamma_{m-1}-1) - C_R(\gamma_{m-1}+\xi-1)) + \sum_{x=\gamma_{x+1}}^{m-2} t_{x-1}^m \frac{x-1}{m} \delta^C_{x,\gamma_x-1}(R_H(\gamma_x-1) - C_R(\gamma_x+\xi-1)). \tag{3.19}$$

We also assume that this is bigger than $C_b - r_{NP}$, since otherwise $\rho_{\gamma_{m-1}} = 0$, which contradicts our assumption that the host will raise. Looking at the value for Γ', we only need consider

$$t_m^m(R_H(\gamma'_{m-1}-1) - C_R(\gamma'_{m-1}+\xi-1)). \tag{3.20}$$

Therefore we get out the same result for $\rho_{\gamma_{m-1}}$ as long as

$$t_m^m(R_H(\gamma_{m-1}-1) - C_R(\gamma_{m-1}+\xi-1)) \geq C_b - r_{NP}. \tag{3.21}$$

Since we have assumed $R_H(\gamma_x - 1) - C_R(\gamma_x + \xi - 1) < 0$, the summation part of (3.19) is also negative, meaning the inequality in (3.21) holds.

2. Now we must perform the induction step.

   Let us suppose that we have found some values of Γ' and that these are identical to the equivalent terms in Γ^* i.e. all the $\gamma'_i = \gamma_i$ for all $i \in (x, m-1)$. We then consider γ'_{x-1} from

$$\Gamma' = [0, 0, \ldots, \gamma'_{x-1}, \gamma_x, \ldots, \gamma_{m-1}]. \tag{3.22}$$

   The new value for $\Pi_{\Gamma'}$ is

$$\Pi_{\Gamma'} = \sum_{y=1}^{m} t_y^m \sum_{z=1}^{y-1} \delta^C_{y-1,z-1}(\rho_z R_p(z-1) - C_{DC}(y-z)). \tag{3.23}$$

Since we know all of the values of $\delta^C_{y-1,z-1}$ we can substitute these in giving

$$\begin{aligned}\Pi_{\Gamma'} \quad &= \sum_{y=x}^{m} t_x^m(\rho_1 R_p(0) - C_{DC}(y-1)) \\ &+ t_x^m(\rho_{\gamma'_x+1} R_p(\gamma'_{x-1}) - C_{DC}(m-\gamma'_{x-1})) \\ &+ \sum_{y=1}^{m} t_x^{y-1}(\rho_1 R_p(0) - C_{DC}(y-1)).\end{aligned} \tag{3.24}$$

We can break this up into the first term, which is the same as in the *true* solution, and the second and third terms, which could (potentially) affect the decision of the host in Stage 7. We again have a situation where we need to check if $\gamma_y = \gamma'_y$. We shall look at the outcome for the host for Γ^*, and again w.l.o.g. we assume $a = 1$ and only look at the part which involves the parasite

$$\begin{aligned}&\sum_{z=x}^{m} t_z^m \frac{z}{m}(R_H(\gamma_{x-1}-1) - C_R(\gamma_{x-1}+\xi-1)) \\ &+ t_x^m \frac{x}{m}(R_H(\gamma_{x-1}-1) - C_R(\gamma_{x-1}+\xi-1)) \\ &+ \sum_{z=0}^{x+1} t_{z+1}^{m+1} \frac{z+1}{m+1}(R_H(\gamma_x-1) - C_R(\gamma_x+\xi-1))\end{aligned} \tag{3.25}$$

which is assumed to be greater than $C_b - r_{NP}$.

We obtain the outcome for Γ' as

$$\begin{aligned}&\sum_{z=x}^{m} t_z^m \frac{z}{m}(R_H(\gamma'_{x-1}-1) - C_R(\gamma'_{x-1}+\xi-1)) \\ &+ t_x^m \frac{x}{m}(R_H(\gamma'_{x-1}-1) - C_R(\gamma'_{x-1}+\xi-1))\end{aligned} \tag{3.26}$$

which is the same as the expression for Γ^* in equation (3.25) except for terms which, under the assumption of the theorem, do not affect the optimal decision.

□

Stage 4b

Now we calculate which is the best choice for the host in *Stage 4*. We know the value of α which is fed in by *Stage 4a*. We assume $a = 1$ then work out the outcome for both host and parasite in the later stages, then assume $b = 1$ and do the same. Finally we compare the expected outcomes for the host against each other and against $-C_A$ (the outcome for $c = 1$) to work out the best choice, which is the one with the largest outcome.

Stage 4a

Given the decision for the host in *Stage 1* and for the parasite in *Stage 2*, we now need to know the expected outcome for both in the chick game. For this we need to work out the outcome for both in the later stages for every possible number of eggs which can reach these later stages. For every $y \in (0, x)$ (where $x - 1$ is the number of host eggs the parasite chooses to leave) we calculate a value for α based upon the equations in the previous section, then use this and feed it into the later games. We then take all these values and work out both $H(n)$ and $P(n, x)$.

Stage 2

Given the value for n from *Stage 1* we just work out which value of x maximizes the outcome for $P(n, x)$.

Stage 1

For this stage we set a sensible maximum for the number of host eggs to lay. Then we calculate the expected outcome $H(n)$ for each n.

3.2 Example Calculations

Stage 6

Since *Stage 7* is just a calculation we can look initially at *Stage 6*. At this stage we have a value for n and α, we assume that all $n = 4$ eggs have made it as well as a parasite with probability $\alpha = 0.1$ (values for n and α chosen arbitrarily but resemble those which occur later in the real-life

Table 3.2: Worked example variables.

Parameter
$R_H(x) = x$
$C_R(x) = 0.25e^{\frac{x}{2}}$
$C_L(x) = \frac{x}{100}$
$C_A = 0$
$C_e = 0.26$
$R_P(x) = e^{\frac{-x}{10}}$
$C_{DA}(x) = \frac{x}{100}$
$C_{DC}(x) = \frac{x}{100}$
$\sigma = 0.68$
$\beta = 0.06$
$s_n^n = 0.99$ and $s_x^n = \frac{0.01}{n} \forall x < n$
$t_n^n = 0.99$ and $t_x^n = \frac{0.01}{n} \forall x < n$

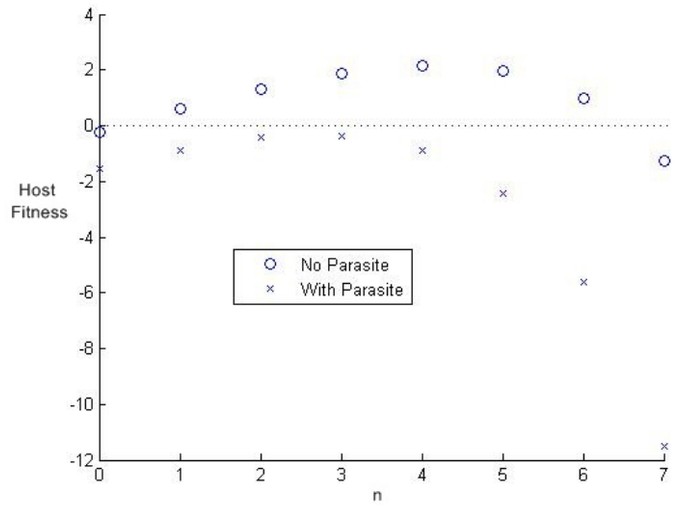


Figure 3.3: Graph of host fitness for a given final number of host chicks for both the cases with and without a parasite chick, i.e. Comparing $R_H(X) - C_R(X)$ with $R_H(X-1) - C_R(X-1+\xi)$ where $R_h(x) = x, C_R(x) = 0.25e^{x/2}, \xi = 2$

examples). We also assume that $a = 1$. The reward functions and other probabilities are shown in Table 3.2

So now we need to work out the best γ for the parasite chick. We start off by looking at

$$\Gamma = [0, 0, 0, 0]$$

meaning that the parasite will destroy all the host eggs in every situation. For this we get the following value for r (The host outcome)

$$r = [-0.0752, 0.0030, 0.0043, 1.9607, 0]$$

where this is the vector containing the values for $r(x)$ for each x from one to five (four hosts and a parasite), the value for five being zero because here the parasite always destroys the host's eggs.

This equates to the following ρ

$$\rho = [0, 1, 1, 1, 0]$$

where we give the value $\rho_y = 1$ if the host will raise and $\rho_y = 0$ if it does not. Thus in this case the host will raise if the nest contains 2,3 or 4 eggs, but not 1; (note that 5 eggs cannot occur here). The outcome for the host is 1.9680 and the parasite's outcome is $-.0397$. In order to compute our best Γ' we need to compare this to the outcome for values of Γ where the entry in the final position is different. We see that the best outcome for the parasite in this case is

$$\Gamma = [0, 0, 0, 0, 3]$$

So we move on and check this against values of Γ with 3 in the final position (as in our definition of Γ', for the different possibilities in the penultimate position. The best outcomes occurs for our original Γ. Note that it appears as if the outcome for the host does not change at all (see Table 3.3). However, this is because of the rarity in which the differing strategies lead to different behaviour in practice, and there are in fact small differences. For examples the strategies $[0, 0, 0, 3, 3]$ and $[0, 0, 0, 2, 3]$ only lead to different behaviours with probability $\alpha \times t_3^4 \times 3/4 = 0.06 \times 0.01 \times 0.75 = 0.00045$ for our example. In fact it turns out that this chosen value of Γ is the best choice overall for the parasite.

Table 3.3: Outcomes of varying elements of γ

Γ	r	ρ	Host Outcome	Parasite Outcome
First Check				
$[0,0,0,0,4]$	$[0, 0.0010, 0.0030, 0.0043, 1.9607, -0.1673]$	$[0,1,1,1,1,0]$	1.9691	0.0079
$[0,0,0,0,3]$	$[0, 0.0010, 0.0030, 0.0043, 1.9160, 0]$	$[0,1,1,1,1,0]$	1.9244	0.7314
$[0,0,0,0,2]$	$[0, 0.0010, 0.0030, -0.0049, 1.9607, 0]$	$[0,1,1,0,1,0]$	1.9648	-0.0119
$[0,0,0,0,1]$	$[0, 0.0010, -0.0237, 0.0043, 1.9607, 0]$	$[0,1,0,1,1,0]$	1.9661	-0.0218
Second Check				
$[0,0,0,3,3]$	$[0, 0.0012, 0.0030, 0.0043, 1.9160, 0]$	$[0,1,1,1,1,0]$	1.9245	0.7309
$[0,0,0,2,3]$	$[0, 0.0012, 0.0030, 0.0043, 1.9160, 0]$	$[0,1,1,1,1,0]$	1.9245	0.7311
$[0,0,0,1,3]$	$[0, 0.0012, 0.0030, 0.0043, 1.9160, 0]$	$[0,1,1,1,1,0]$	1.9245	0.7312

Table 3.4: Outcome for different Stage 4 Decisions

Stage 4 Decision	Best Γ	Host Outcome	Parasite Outcome
a	$[0,0,0,0,3]$	1.9244	0.7314
b	$[0,0,0,0]$	1.7468	0.9583
c	NA	0	0

Stage 4b

Suppose that we again assume that $\alpha = 0.1$. We need to work out which is the best choice at Stage 4, and so we need to find the outcome for a, b, or c.

It is clear from Table 3.4 that the host will choose a in this case. It is worth noting that the parasite reward for $b = 1$ is the largest of the three possibilities in this example, which is initially surprising as this is when the host attempts to remove the parasite by ejecting a single egg. The reason for this is that the parasite only records this outcome if the host chooses to eject, guesses incorrectly and destroys one of its own, meaning the parasite will have less destruction to do. In reality the parasite will receive $1 - \sigma$ times this reward, but this is not calculated until Stage 2.

Stage 4a

Here we calculate the value of α going into this second half. For example assuming $n = 4$ and that the parasite adult does not choose to destroy any eggs, we get

$$\mathcal{A} = [0.0299, 0.0442, 0.0581, 0.0002, 1.0000]$$

where A is a vector where the entries are the probabilities that there is a parasite given different values of $m \in (0, 5)$. In this case $a = 1$ and the outcome for the host is 2.1403 and for the parasite

Table 3.5: Outcome for different Stage 2 Decisions when $n = 4$

δ decision	Parasite Outcome
δ_0^A	-0.0350
δ_1^A	0.8533
δ_2^A	0.7811
δ_3^A	0.7162
δ_4^A	0.3032

Table 3.6: Outcome for different Stage 1 Decisions

n	Host Outcome
1	0.4237
2	1.1694
3	1.6824
4	1.9187
5	1.7380
6	1.4959

is 0.9583, with the chosen Γ being the decision for the chick to destroy everything.

Stage 2

As an example we assume in this case that the number of eggs laid is 4, so we need to look at the parasite outcome for the different δ^A's, as we can see in Table 3.5. Thus the parasite decides to leave just one host egg.

Stage 1

Choosing 6 as a maximum for n in this example, we just look at the outcome for each of the possible n (see Table 3.6). This gives us $n = 4$ as our best choice for the host.

3.3 Results

In this section we describe two real interactions between a host and its parasite. In each case we use real parameter values as much as we can and make use of other evidence to estimate further parameters indirectly. These then generate predictions of behaviour for the two cases. We further consider varying a range of parameters to allow for different estimates and examine the effect. We will look at two interactions between host and parasite, the first the Yellow Warbler (host) and the Brown-Headed Cowbird (parasite), the second the Reed Warbler (host) and the Common Cuckoo (parasite).

Table 3.7: Real World Example Variable Table.

Parameter
$R_H(x) = x$
$C_R(x) = 0.25e^{\frac{x}{2}}$
$C_L(x) = \frac{x}{100}$
$C_A = 0$
$C_e = 0.26$
$R_{cuckoo}(x) = e^{-0.1x}$
$R_{cowbird}(x) = \frac{50-(x-2.25)^2}{50}$
$C_{DA}(x) = \frac{x}{100}$
$C_{DC}(x) = \frac{x}{100}$
$\sigma_{cuckoo} = 0.68$
$\sigma_{cowbird} = 0.98$
$\beta_{yellow} = 0.64$ $\beta_{reed} = 0.06$
$s_n^n = 0.99$ and $s_x^n = \frac{0.01}{n} \forall x \neq n$
$t_n^n = 0.99$ and $t_x^n = \frac{0.01}{n} \forall x \neq n$

3.3.1 Yellow Warbler vs Brown-Headed Cowbird

This is an interesting interaction because the Brown-Headed Cowbird is a species that does not generally eject any host chicks after hatching, however on occasions they have been seen doing so [9], so it is clear that they are capable of it. Thus although cowbirds do not (usually) in reality destroy chicks in this situation, our model allows them the option to do so. Parasitism occurs for the Yellow Warbler in a high percentage (64%) in [54], and so we choose $\beta = 0.64$. Other studies [1, 2] show similar statistics. The Yellow Warbler makes correct guesses as to which egg in the nest is the parasite (if it chooses to eject) 98% [54] of the time, so we choose $\sigma = 0.98$. Which may seem high and strikes the question as to why the warbler doesnt always eject. But we see that the effect of doing so is more detrimental than the profit from ejecting.

From studies of the warbler/cowbird interaction it is shown that it is approximately (2-2.5) [9] times harder to raise a cowbird chick than a warbler chick; we shall use $\xi = 2.25$. According to studies done by [9] the reed warbler host loses an average of 0.26 [9] of its own eggs during ejection; since there is little data on this on the yellow warbler we shall assume it is the same. We usually set the fitness to be the average number of host eggs left at the final stage so we shall use this as our cost of ejection C_E. We also assume in this case that the cost of abandonment C_A is equal to zero.

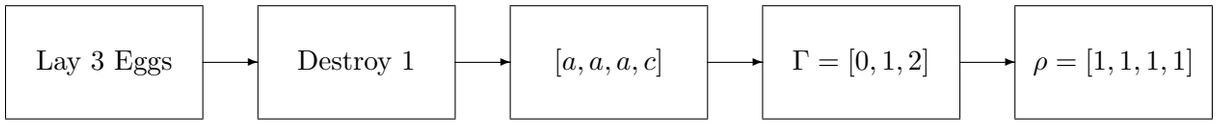


Figure 3.4: Stages of the game for the cowbird

We also need suitable values for our fitness parameters. First of all we look at the reward to the host. We always set $R_H(x) = x$, which makes sense because the fitness is just the number of eggs we get out minus the cost it took to raise them. In this case we can set it as an arbitrary value $0.1x$. Therefore it costs 10% of the reward from a host chick surviving to fledge to raise it.

It has been shown in studies that a parasite does best with approximately 2-2.5 host chicks in the nest [23]. For this reason we shall in this first example make the payoff graph for the parasite the following.

$$R_P(x) = \frac{50 - (x - 2.25)^2}{50} \tag{3.27}$$

Clearly equation (3.27) has a maximum at 2.25. The destruction costs for both the adult cowbird and the cowbird chick are set at 0.01 per host chick destroyed (just a small nominal cost). It has not proved possible to find experimental evidence for an explicit functional form for the fitness cost $C_R(x)$ to the host in raising a clutch. We choose a form that has plausible features, namely a small cost for small clutches and an increasing incremental cost for each extra egg for larger clutches. Different forms from the one chosen are possible, but as long as they maintain these general features, then we contend that the results would not be greatly affected. We get the following outcome for the cowbird game

- $\Omega = 1.1476$
- $\Pi = 0.9678$

where the stages are as described in Figure 3.4.

This solution means that the host will lay three eggs; if a parasite visits the nest it will destroy one of the host eggs and lay one of its own (it does this because as we can see in Figure 3.4 the host will abandon the nest should it see four in the nest). These results follow that of [54], where evidence of one egg being removed by the parasite adult was found. Once the chicks have hatched, no matter what has happened with natural destruction, the parasite will not destroy any of the

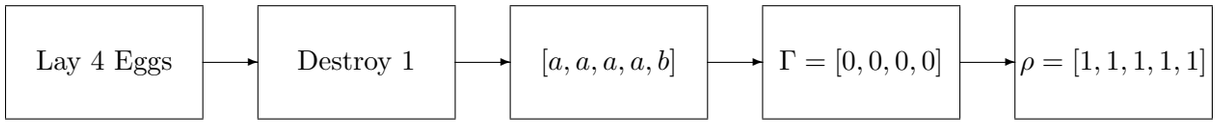


Figure 3.5: Stages of the game for the cuckoo

host chicks and the host will then raise the brood whatever the number in the nest. This is what we find happens in nature with real cowbirds.

3.3.2 Reed Warbler vs Common Cuckoo

Note that the Common Cuckoo has a very different behaviour from Cowbirds [22, 14] in that it destroys all of the host chicks [6, 9]. We assume that all the natural elements are the same for this game as for the one with the Cowbird and Yellow Warbler, including the rewards and costs to the host, except in the case of the parasitism rate, which is much lower here (6%) [9]. The only things that we are going to change are the fitness equation for the cuckoo and the raising cost to the host of the parasite chick ξ. Unlike for the cowbird, there is no evidence that the cuckoo would benefit from the presence of host young since it has always been seen to destroy the host eggs, so we set the value of $R_P(x)$ as in equation (3.28).

$$R_p(x) = exp(-0.1x) \tag{3.28}$$

As shown in Krüger and Davies [24] a common cuckoo bird is over four times the size of its hosts, so we set $\xi = 4.377$, the average value found. Cuckoos are better mimics than Cowbirds and it has been shown that the ejection success of the Warbler versus the Cuckoo is only 68% [9], so we set $\sigma = 0.68$. We get the following outcome, with the game described in Figure 3.5.

- $\Omega = 1.5784$
- $\Pi = 0.7162$

We initially have the same story happening as with the Cowbird, where the host will lay four eggs and if a parasite visits the nest it will destroy one of the host eggs and lay one of its own. However after the chicks hatch behaviour is different, as the Cuckoo chick will destroy all of the host's young no matter how many there are left in the nest. This is again the behaviour of real cuckoos. It should be noted that we can obtain the type of behaviour associated with the cowbird,

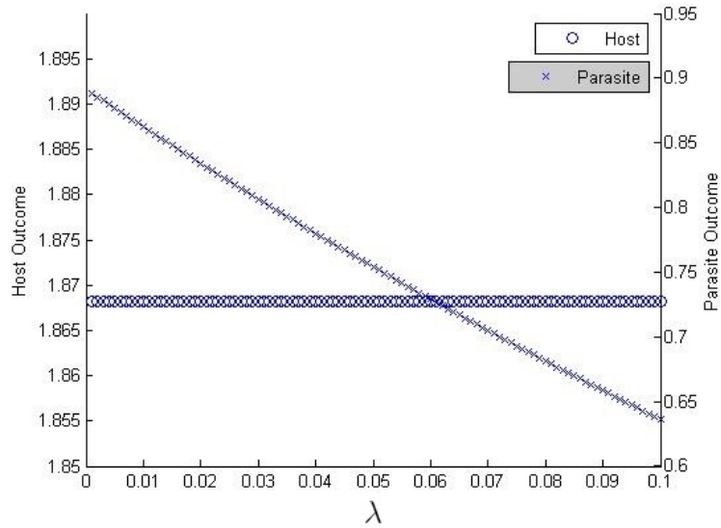


Figure 3.6: The parasite reward function $R_P(x) = e^{-\lambda x}$ with $\xi = 3$. Other parameters are $\beta = 0.06, \sigma = 0.68, R_H(x) = x, C_R(x) = 0.25e^{x/2}, C_{DC}(x) = C_{DA}(x) = 0.05x$

described above, with the same exponential shape of reward as in the cuckoo, providing that the rate of decay of $R_P(x)$ is sufficiently slow. Meaning that if the reward of having other hosts doesnt get too high it wont proceed with destroying and will act in a similar way to the cowbird.

3.4 Differing Parameters

3.4.1 The parasite reward $R_p(x)$

In Figure 3.6 the values of C_{DA} and C_{DC} have been increased from their default values to consider a situation where behaviour varies for plausible values of λ the tolerance of a parasite to having host chicks in the nest with it. The higher the value of λ, the worse for the parasite it is to have host chicks being raised alongside it. The pattern of the outcomes is the same, except that these occur for larger values of λ in this figure than they would if we had used the default values. The reward for the parasite steadily decreases, whereas the host reward marginally increases but as we can see from the scale, this reward is not changed a lot. In fact from the figure it is not clear that there is any strategic change at all, as there are no significant jumps in the rewards to parasite or host; however such strategic changes do occur. There are always four host eggs laid, and if a parasite visits, it will always eject a single host egg. In the region between $\lambda = 0.05$ and $\lambda = 0.06$, there are in fact three points where a decision change has been made. These occur at roughly $\lambda = 0.0515$, $\lambda = 0.053$ and $\lambda = 0.055$. This is a transitional period between typical cowbird behaviour (low λ)

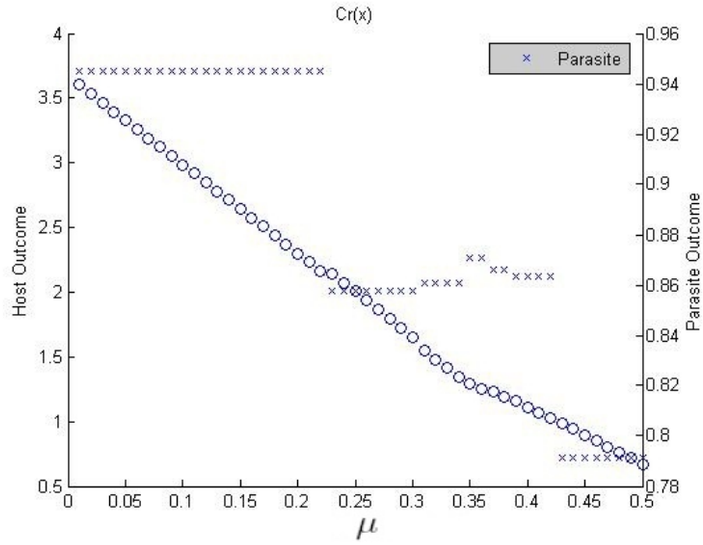


Figure 3.7: The cost function to the host of raising Cuckoo chicks $C_R(x) = \mu e^{\frac{\mu s}{2}}$. Other parameters are $\beta = 0.06, \sigma = 0.68, R_H(x) = x, R_P(x) = e^{-0.1x}, C_{DC}(x) = C_{DA}(x) = 0.1$,$\xi = 4.377$

and typical cuckoo behaviour (high λ). For values less than $\lambda = 0.0515$ we get that the parasite ejects one egg in Stage 2 but does not eject any in Stage 6. For $\lambda \in (0.0515, 0.053)$ the parasite will eject in Stage 6 if it has only a single nest-mate (the others being lost through natural destruction). For $\lambda \in (0.053, 0.055)$ the parasite will eject all in Stage 6 if there are only one or two others. For any value of λ higher than 0.055 the parasite will eject all three of the host chicks.

3.4.2 The raising cost of the host $C_R(x)$

The value of μ in Figure 3.7 relates to the cost to the host of raising a chick; the higher the value of μ, the greater this cost is. The outcome for the host differs greatly depending on the cost of raising, as we would expect. However there is a change in the parasite's outcome which is not necessarily as we would expect, since this does not have a direct relation to $C_R(x)$. This reward is not smooth and jumps at certain points, these being caused by a change in the host's behaviour. When the value of μ reaches 0.35 the host then chooses to only lay three eggs which is why we see a slight raise in the parasite outcome which then slowly dies away.

3.4.3 The probability that the host correctly rejects the parasite egg σ

Figure 3.8 shows the change in outcome for the host and a cuckoo parasite; we can see that the host does better when σ is high and the parasite does better when σ is low, as we would expect. There is in fact only one change in possible decisions, when the parasite adult performs the destruction

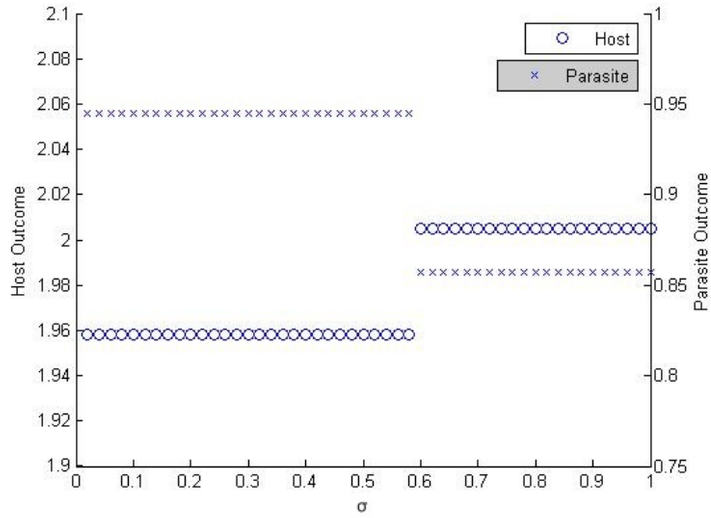


Figure 3.8: The probability that the host correctly recognises the Cuckoo chick σ. Other parameter values are $\beta = 0.06, R_P(x) = e^{-0.1x}, R_H(x) = x, C_R(x) = 0.25e^{x/2}, C_{DC}(x) = C_{DA}(x) = 0.1$,$\xi = 4.377$

for low values of σ and the parasite chick destroys the host chicks for high values of σ. There is no change in outcome for host and cowbird in their game when we change σ, so we have omitted the graph. In this case, the host never tries to evict the cowbird parasite, because it is tolerant of the host's young.

3.4.4 The relative cost of raising a parasite chick ξ

Here Figure 3.9 breaks down into different points where the parasite's decision changes as it takes into account its own value for ξ, and the host's potential reaction. For the cuckoo example, behaviour is as follows for a different ξ.

0-0.5 For small values of ξ the adult parasite chooses to destroy all the host eggs. At Stage 4, the probability that there is a parasite given that the parasite would choose to destroy them all is 0.65. The expected outcome for the host (with $\xi = 0.2$) is 0.45, so is still positive. The host will abandon a single chick in Stage 5 if ξ goes above 0.5.

0.5-1.8 For these values of ξ the host will still allow the parasite chick through at all times except when there is just one chick. Most of the destruction this time is done by the parasite chick, with the adult destroying one egg to leave the nest the same size as when the host laid it, thus lowering the chance that the host believes there is a parasite.

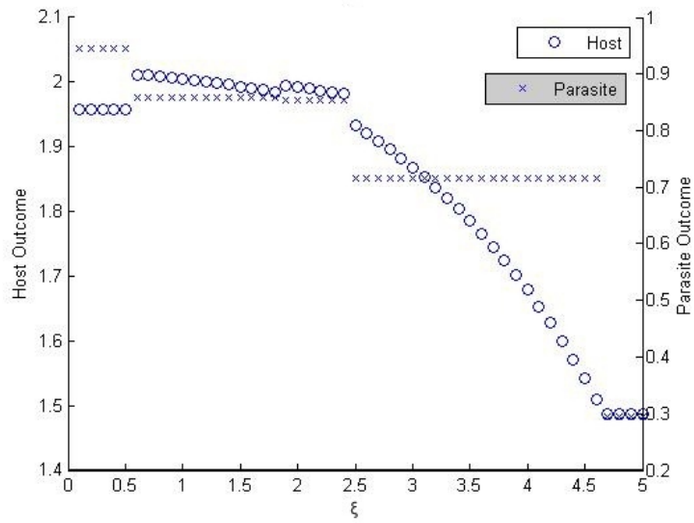


Figure 3.9: The demand on resources of a parasite chick ξ for the Cuckoo. Other parameter values are $\beta = 0.06, \sigma = 0.68, R_H(x) = x, C_R(x) = 0.25e^{x/2}, R_P(x) = e^{-0.1x}, C_{DC}(x) = C_{DA}(x) = 0.1$

1.8-2.5 The host will still abandon a single egg at Stage 4. The parasite adult ejects down to one host egg in addition to its own egg, increasing its chances of being raised.

2.5-4.5 The host will now abandon at all points unless there are the same number of eggs in the nest as it first laid, so the parasite just destroys one.

4.5+ Here the parasite strategies for adult and chick do not differ from the 2.5-4.5 range. It turns out that whatever its decision as an adult the host will attempt to destroy it by ejecting (or in some cases abandoning) in Stage 4. Thus the parasite must rely on luck, where the host fails to correctly identify it, in order to survive.

With the cowbird example there is only one change in decision which occurs at about $\xi = 4.5$ as with the cuckoo, where before this time the parasite will destroy one host as an adult then the chick will leave the nest alone. Beyond this the parasite adult and the host make the same decision as decribed in the 4.5+ range for the cuckoo. However, the chick decision is different choosing not to destroy at any point.

3.4.5 Parasite frequency β

Varying β to see if the different values alter the decision is of special interest because it does appear that in nature strategies do vary depending upon the level of parasitism [3], and this is a parameter for which reliable estimates can be found. We revert to the default value of $R_p(x)$, C_{DA} and C_{DC}.

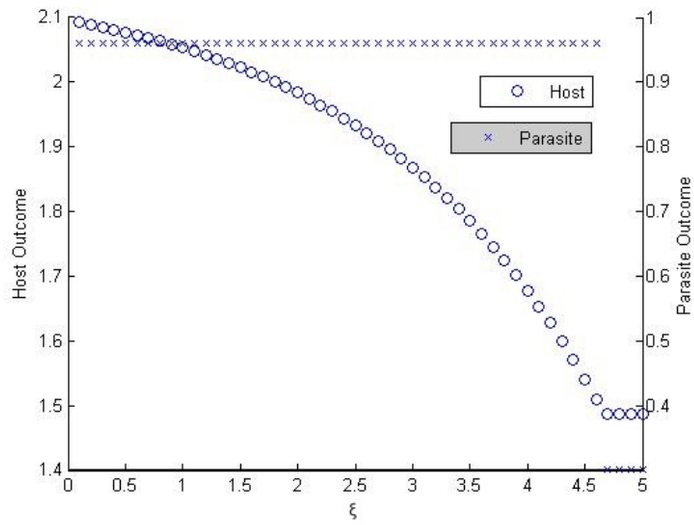


Figure 3.10: The demand on resources of a parasite chick ξ for the Cowbird. Other parameter values are $\beta = 0.06, \sigma = 0.98, R_H(x) = x, C_R(x) = 0.25e^{x/2}, R_P(x) = e^{-0.1x}, C_{DC}(x) = C_{DA}(x) = 0.01x$

For the varying values of β we get the different outcome for the hosts as shown in Figure 3.11. This shows that as we would expect, the outcome for the host will decrease as the probability of a parasite arriving increases except for the slight increase at a change of strategy. This is most likely due to the discreet nature in which the graphs are calculated. The outcome of the parasite is independent of β, except for the effect of varying host strategy; this occurs once, with a significant reduction in the parasite outcome when β increases beyond a critical value. At below this critical value the parasite behaves the same as for low values of ξ where the adult destroys all of the host eggs.

Looking at the change in β for the cowbird in Figure 3.12, we see three distinct changes in outcome. For small β we get a similar outcome for the host, but where it lays four eggs the Cowbird will eject one and lay one of its own, and the chick will not destroy. In the middle section the host will only lay three as described above and once β becomes high it will lay four (its preferred number in the absence of parasitism) and then attempt to destroy the parasite in Stage 4 no matter what the parasite adult chooses to do. This is because of the high probability of it being able to identify the cowbird and the high probability of there being one visiting the nest.

This model only accounts for one single interaction in one single breeding season. Having a high parasitism rate one year could effect the rate the next year as many models have shown. Takasu [52] for one.

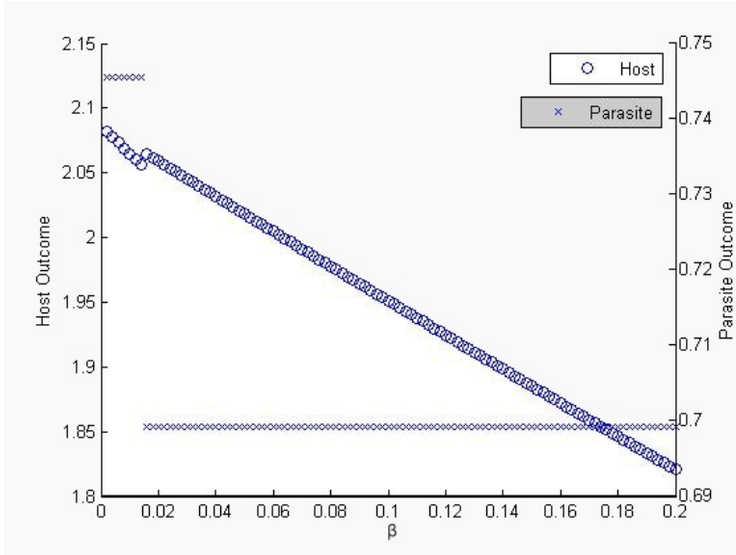


Figure 3.11: The probability that a random nest is visited by a Cuckoo β. Other parameter values are $R_P(x) = e^{-0.1x}, \sigma = 0.68, R_H(x) = x, C_R(x) = 0.25e^{x/2}, C_{DC}(x) = C_{DA}(x) = 0.01x$,$\xi = 4.377$

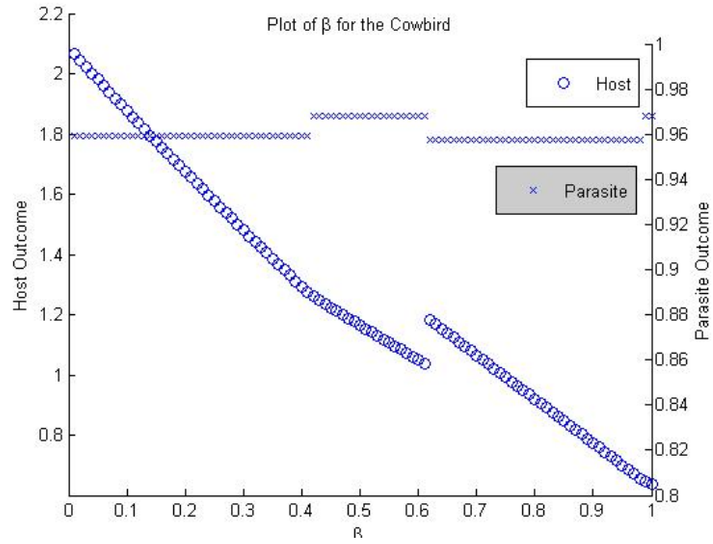


Figure 3.12: The probability that a random nest is visited by a Cowbird β. Other parameter values are $R_P(x) = e^{-0.1x}, \sigma = 0.98, R_H(x) = x, C_R(x) = 0.25e^{x/2}, C_{DC}(x) = C_{DA}(x) = 0.01x$,$\xi = 2.25$

3.4.6 Particularly significant variables

When considering which of these changes of variables are the most significant, probably the most important thing is looking at variables which when altered produce a change of behaviour. For the parasite the most significant feature, unsurprisingly, is $R_P(x)$ (i.e. its own reward with regards to how many host chicks are in the nest). Changes in this function produce the change in decisions between the two classic behaviours, that of a cuckoo and that of a cowbird.

For the host perhaps the most significant parameter is ξ, the effective cost of raising a parasite, and all other things being equal this is the one variable that affects the host decision in the most ways (so there can be a sequence of different host strategies as ξ varies), although there is also a parasite reaction to such host changes which affects the host's outcome. As ξ increases the cost to the host of raising such a chick increases and the host becomes more aggressive with how it deals with a parasite egg, eventually trying to destroy it no matter the consequences.

Another parameter that has a significant effect on both host and parasite is σ, the probability of correctly identifying a parasite egg. The host will never attempt to evict if this probability is sufficiently low, and evicts if the likelihood of a parasite being present is sufficiently large for larger values of σ. Although this is a single change in behaviour, the outcomes for both parasite and host change markedly when this takes place.

3.5 Discussion

We have developed a model of the interaction between a brood parasite and its host which is based upon a sequence of events, representing decisions by parasite parent, parasite chick and host parent, and random acts of nature. In this way we try to use information about the sequence of real occurrences during these complex interactions in order to improve on past models and give realistic predictions of host and parasite behaviour. Although we consider a very specific sequence of interactions, the methodology is quite general and different sequences of events could be modelled in a similar way without many modifications.

The model is complex in that it has many different elements to it and the interaction between the different parameters can in some instances be difficult to interpret. However it has been shown that there is some significant alteration to the outcome of the game when the parameters are changed and each of the seven stages gives us an interaction between host and parasite that occurs

in nature. We have attempted to identify the key influences of the parameters by concentrating on each in turn, although it should be noted that with this number of parameters it is very hard to identify all the possible interactions between them. A major aim of this section is to consider the sequence of events in detail, and of course in some ways the model is still a simplification. Thus every element of the game has a significant influence, and thus is of potential importance.

A complication of this game is a lack of complete information. In particular, the host is unsure about whether there is a parasite present in the nest or not, and must rely on estimating the likelihood of a parasite based upon the current state of the nest. This lack of complete information in a game in extensive form makes it complicated to analyze [55]). This is a simplification of reality, and sometimes a host can pick up cues as to whether a parasite is present or not. For instance if the nest is disturbed by the parasite, or if the parasite egg is sufficiently different from those of the host that the host can recognise this.

The number of possible sequences of games that can occur quickly becomes very large as the initial number of eggs laid increases. In particular the vector Γ, which describes the possible choices of the parasite chick for all possible numbers of host eggs in the nest, can have $(n+1)!$ possibilities and the computing time involved in running the program this number of times is prohibitive. In fact, we use a simpler procedure which is much quicker. We have proved that the solutions obtained for the two methods are in fact the same for the game described under clearly defined conditions.

This model assumes that both players in the game know all of the rules. It also assumes that both the host and the parasite know what the other would do given what they themselves do. For example when the parasite in *Stage 2* is choosing whether to leave four eggs and itself, then it knows what the host will do in *Stage 4* if it sees five eggs. At the final stage when the host decides whether to raise or not, it knows the number of eggs present m and has an estimate of the probability that one of these is a parasite α, acquired from information from the previous stages.

We assume in this model that only one parasite visits each nest and only lays one egg. This is to make the calculation simpler. But it is entirely plausible to include more parasites having multiple *Stage 2*s and *Stage 6*s. To adjust the model, we would need to consider how the parasites interact. Earlier models, in particular Maruyama and Seno [29], have considered the important question *How many eggs should the Parasite lay?*. This will be looked at in more detail in the next section.

Increasing the prevalence of parasitism through the parameter β has an effect on both host and

parasite. Common sense would suggest that the effect on the host would be the more significant, as the number of interactions between host and parasite is proportional to this parameter, and such interactions are detrimental to the host, whereas our model assumes no interactions between parasites at all. However, in fact, the effect on the host is quite gradual, whereas the effect on the parasite reward occurs in a sudden steep jump as the host changes strategy in response to the frequency of parasites, the reward becoming less as the parasite frequency increases. See Broom et al [4] for similar results. The effect of this parameter is of interest because it is measurable and so predictions are potentially testable, and it would be of interest whether the sequence of events that occurs in the model as β changes also happen for real situations. Similarly the cost of raising a brood should affect the host but not the parasite, but in fact a steady decline for the host is accompanied again by steep (negative) jumps for the parasite, as the host changes strategy. In this model we have considered a fixed value of β only, rather than allowing it to vary as would happen if we considered a fully population dynamic model. Our approach has been to assume that the population has settled to a stable situation (or at least that this rate of change is slow enough that birds' strategies are able to change quicker than the rate of parasitism) and look for ESSs. There are likely to be situations where this is not the case, and then a more dynamic model would be valuable.

It would be interesting to test some extreme examples within this model. Such as a brand new parasite acting on an unknowledgable host or a cuckoo type invading the nest of the yellow warbler which is usually parasisited by Cowbirds. The co-evolution theory suggest that in examples both parties will recieve a reduction in reward. Since the maximal stable strategy cannot be invaded by an outsider.

Please note that this chapter has been published as a paper in the Journal of Theoretical Biology. [15].

Chapter 4

Single Interaction Multiple Parasite Model

4.1 Simplifying the Single Interaction Model

The problem of computational speed was helped by the theorem in the previous chapter. However, if we wish to expand this model, we must lower the amount of computations still further. The way we shall do this is to swap two of the stages that happen at the same time (at least for the host), these being Stages 5 and 6, the natural destruction of chicks and the parasite destruction of chicks. In the previous model we had the nest hatching with some number of the brood and parasite being destroyed. The parasite (if it survived) would then make a decision based upon the number of eggs in the nest at that time. This seems the most sensible of scenarios. However, if we assume no destruction happens in the majority of the interactions then having the parasite destruction stage go before natural destruction gives the same result. The advantage of doing this is that the parasite makes a single decision based upon $m-1$ (the total number of host chicks) as opposed to several possible decisions based upon what has happened in natural destruction. Note that this will not affect the adult game at all. First we look at how this affects the reward functions for the host and parasite.

Obviously this will change the reward function for the host in Stage 7. Now we assume that m eggs survived to Stage 5 and that the parasite's decision is to destroy down to $x-1$ hosts eggs, leaving a total of x. Then we look at the possible ways for y eggs to remain at the end and find the value of $r(y)$.

We get

$$(1-\alpha)t_y^m(R_H(y)-C_R(y))+\alpha\left[t_y^x\left(\frac{y}{x}(R_H(y-1)-C_R(y-1+\xi))+(1-\frac{y}{x})(R_H(y)-C_R(y))\right)\right]\geq C_b. \tag{4.1}$$

If we denote $R_{NP}(y)=R_H(y)-C_R(y)$ and $R_P(y)=R_H(y-1)-C_R(y-1+\xi)$, we can write (4.1) as

$$(1-\alpha)t_y^m R_{NP}(y)+\alpha\left[t_y^x\left(\frac{y}{x}R_P(y)+\left(1-\frac{y}{x}\right)R_{NP}(y)\right)\right]\geq C_b. \tag{4.2}$$

Probability of a Parasite (α)

We can rearrange (4.2) to get a critical value for α, meaning the host raises if

$$\alpha\leq\frac{t_y^m R_{NP}(y)-C_b}{\left[t_y^m-\left(1-\frac{y}{x}\right)t_y^x\right]R_{NP}(y)-t_y^x\frac{y}{x}R_P(y)} \tag{4.3}$$

Let us look at an example of how this works in practice. We define the following

- $m=4$,
- $y=4$,
- $x=4$ (3 hosts and the parasite),
- $R_{NP}=3$,
- $R_P=-1$,
- $t_4^4=0.99$,
- $C_b=0$.

These are arbitrary values for each parameter and they resemble those we got from the Cowbird vs Yellow Warbler example from the previous section. From this we get

$$\alpha\leq\frac{0.99\times 3}{0.99*3+0.99}=0.75.$$

Meaning the host will raise as long as $\alpha\leq 75\%$.

4.2 The Multiple Parasite Model

We can now extend the model to include the possibility of more than one parasite arriving at the nest and laying an egg. This has been shown to occur is some studies such as Gaston [13], McLaren et al [31] and De Marsico and Reboreda [10]. It also makes logical sense to expand this model to allow for multiple parasites to see the if the interaction between parasites has any effect on the interaction between the hosts and parasites. One would believe that this competition will prove to show an improved outcome for the host. However, if the parasites show a more aggressive parasitism behaviour due to increased competition then this could work out worse for the hosts. Not to mention this brings in the possibility of more than one parasite being raised in the nest, increasing the costs to hosts. If the probability of parasitism is high the chances of only singular parasitism becomes very low; in the cases we looked at previously where we only have a 6% chance of parasitism then the chances of multiple parasitism is low (0.0036 for double parasitism) but if this goes up to just 25% this becomes a more significant 5% chance of double parasitism. We can now explain the possible stages as follows.

S1 This is again the first and overall main decision by the host, which is the choice of how many eggs (n_h) to lay at the beginning of the game.

S2 We assume that parasites will arrive at a nest during the laying period at a constant rate with a mean number of arrivals λ_p. Thus the probability of n_p parasites arriving (we shall call this $\mu_{\lambda_p}(n_p)$) is given by:

$$\mu_{\lambda_p}(n_p) = \frac{e^{-\lambda_p}\lambda_p^{n_p}}{n_p!}$$

Stage 2 is again the decision by the parasite to destroy a number of the host eggs. But also there is now the possibility that it can destroy some of the parasites that have preceded it. We set the decision of how many to destroy down to now as δ_x^A which means the parasite will destroy down to x no matter how many eggs are in the nest.

S3 This is the first of two natural destruction stages, and it affects both host eggs and the parasite eggs (if there are such eggs). If there are x host eggs in the nest and no parasite then the probability that y host eggs survive is again given as s_y^x. If y eggs are left in total at this point then we assume that the individual parasites has a probability of survival of $\frac{y}{x}$ (i.e. each parasite has the same chance of survival as each host egg). If we assume that x

eggs end Stage 2 with x_p of them being parasites (x_h host eggs), the number of remaining eggs follows a multinomial distribution; if y eggs are left after Stage 3 the probability that y_p parasites have survived is

$$\frac{\binom{x_p}{y_p}\binom{x-x_p}{y-y_p}}{\binom{x}{y}}. \tag{4.4}$$

S4 This is again a decision that occurs before hatching by the host. Since we can now have any number of parasites in the nest it can make a decision of ejecting more than one egg, to try to get rid of the parasite eggs. We have the same outcome as we did before where the host chooses either a, b or c which in turn refers to not do anything and allow the nest to continue without ejecting, attempt to eject the parasites or abandon the nest.

We assume a probability σ which is the success rate that the host will have at ejecting eggs. If the host chooses b then with probability σ it will eject all the parasites in the nest and with probability $1-\sigma$ it will eject some if its own eggs. Since there are not a given number of parasite eggs, we cannot simply remove one of the hosts eggs as we did before. We must choose a number of eggs to eject, we choose the amount of eggs the host will eject as the number of parasite eggs that are most likely to be in the nest at this time. For instance if $y_p = 2$ gives the largest value in (4.4) then the host will eject two of its own eggs. Note that it will always eject at least one of its own eggs. There is no simple way of calculating the number the host will destroy. Taking the most likely number of parasites in the nest is the most convenient way to do this mathematically.

S5 We must make an assumption as to which parasite makes the destruction decision. We assume this is a decision by the first parasite chick to hatch. We use the term $\delta^A_{m-1,x-1}$ to define the decision to destroy $m-x$ eggs so leaving $x-1$ host and other parasite eggs (so x eggs in total). If it does, then as before, it will incur the cost $C_{DC}(x-y)$. We assume the first parasite to hatch is the only one to make decisions. The other parasites will still receive a reward if they survive. Gaston [13] mentions that nests containing more than one parasite reared them all in 50% of his studies, so there are opportunities where the multiple parasites will survive. In Cuckoo-types however we would assume that if it destroys all hosts it would also destroy all hosts and the parasite(s).

S6 This is the second natural destruction stage where the probability of x being destroyed down to y being t_y^x as before.

S7 This is the final decision of the host whether to raise the full brood or not. Supposing that the number of eggs that have survived to this stage is y. The fitness cost of raising the parasites may be higher for a host parent than the cost of raising a chick of its own, this extra cost being denoted by $y_p\xi$, so that the cost of the parasite chick is ξ times more than that of a host chick and where y_p is the number of parasites in the nest. Thus if there are y_p parasites the cost to the host becomes $C_R(y - y_p + y_p\xi)$ and if it has no parasites in the nest then it will incur a cost of $C_R(y)$.

The parasite will receive a reward depending on how many host eggs there are in the nest. This is denoted as $R_P(y-1)$. The host will receive a reward depending on how many of its own eggs make it to this stage, denoted $R_H(y)$. The host's decision will be denoted by ρ_y, the probability that given there are y chicks in the nest at this final stage, the host will raise them and again, in most cases this probability will either be 1 or 0.

4.2.1 The Chick Game

As before we break this game down into two smaller connected games, the *Adult game* and the *Chick Game*, one dealing with the parasite adult(s) and one dealing with the chick(s). Considering the chick game, assume that there are m chicks that have reached this stage and that m_h of them are hosts and m_p of them are parasites. The first parasite makes a decision of how many of them are to be destroyed in this case. We get the value of $r(x,m)$ (the expected reward to the host if x eggs are in the nest given that m made it to Stage 5) as follows.

Stage 7

$$r(x,m) = \sum_{m_p=0}^{m} \alpha_{m_p} \sum_{x=1}^{m-1} \delta^C_{m-1,x-1} \sum_{x_h=0}^{x} M(m_h,x_h) \sum_{y=0}^{x} t_y^x \sum_{y_h=0}^{y} R(x_h,y_h), \tag{4.5}$$

where

$$M(m_h,x_h) = \begin{cases} \frac{\binom{m_h}{x_h}\binom{m-m_h-1}{x-x_h-1}}{\binom{m-1}{x-1}} & if\ \ m_h \geq x_h\ and\quad m-m_h \geq x-x_h, \\ 0 & \text{otherwise.} \end{cases}$$

$$R(x_h, y_h) = \begin{cases} \frac{\binom{x_h}{y_h}\binom{x-x_h}{y-y_h}}{\binom{x}{y}}(R_H(y_h) - C_R(y_h + (y - y_h)\xi)) & if \;\; x_h \geq y_h \; and \quad x - x_h \geq y - y_h, \\ 0 & \text{otherwise.} \end{cases}$$

Equation (4.5) breaks down in the following way. α_{m_p} is the probability that m_p of the m eggs of are parasites. Then the first parasite (if of course there is one) destroys down to x including itself, of which x_h are hosts. The next summation is dealing with the natural destruction where the x eggs are destroyed down to y with the x_h hosts reduced to y_h and the host getting the reward for raising those y_h chicks.

Note that $M(m_h, x_h)$ is defined only in the areas where the value of m_h is no bigger than x_h meaning than there cannot be more host eggs in the later stages that there were previously which the summation would otherwise allow and we have a similar sequence of events for the parasites. The only possibilities we need to look at are those in which we have a number of parasites and hosts which are no bigger than those which originally survived to this game. For example if we have five eggs with two of them being parasites and they are destroyed down to 3 by the first parasite ($x = 3$) then we do not need to look at the possibility that $x - x_h = 3$ (i.e. 3 parasites). $R(x_h, y_h)$ is similarly defined so that there cannot be more hosts or parasites than there were before so we discount this possibility by setting the reward to zero for those values.

Stage 5

We now look at the different rewards for the parasites. We have different rewards for the first parasite to hatch from that of the other parasites,

$$\Pi^{C1}_{x-1} = \sum_{y=1}^{m} t^{x-1}_{y-1} \frac{y-1}{x-1} \rho_y R_P(y-1) - C_{DC}(m-x). \tag{4.6}$$

For the other parasites there are no decisions made so we get

$$\Pi^{C2}_{x-1} = \sum_{x=1}^{m} \delta^{C}_{m-1,x-1} \frac{x-1}{m-1} \sum_{y=1}^{x} t^{x-1}_{y-1} \frac{y-1}{x-1} \rho_y R_P(y-1). \tag{4.7}$$

The other parasites have the advantage of not having a cost of destruction but may also get destroyed themselves either as shown in the first summation by the parasite in Stage 5 or in the second summation by nature in Stage 6. We assume in this case that there is no extra effect of the other hosts (i.e. using ξ in the R_P function). This has been done for several reasons; firstly, it has

not been shown to make a difference in the decision in nature; secondly, it adds in the complications we took away in simplifying the model and thirdly it does not affect the host-parasite interaction directly (which is the main aspect of the game we wish to study).

4.2.2 The Adult Game

The complications with this game come with the adult sub-section, which is due to the fact that we have more than one parasite that we must account for in Stage 2. For this reason we will need to look at several new aspects to the game.

The probability that given n_p parasites arrive at the nest there are x_p still there after Stage 2

At first sight this may seem to be just a use of the binomial distribution, however since we have parasites arriving in sequence the probability is different. For example consider the possibility that $n_p = 5$ and $x_p = 3$, meaning we have 3 parasite eggs left after Stage 2 given 5 parasite adults arrive. We shall refer to this probability as $\phi(n_p, x_p)$.

$$\begin{aligned}
\phi(5,3) &= \frac{1}{x}\frac{1}{x}\frac{x-1}{x}\frac{x-2}{x} \qquad (4.8)\\
&+ \frac{1}{x}\frac{x-1}{x}\frac{2}{x}\frac{x-2}{x}\\
&+ \frac{1}{x}\frac{x-1}{x}\frac{x-2}{x}\frac{3}{x}\\
&+ \frac{x-1}{x}\frac{2}{x}\frac{2}{x}\frac{x-2}{x}\\
&+ \frac{x-1}{x}\frac{2}{x}\frac{x-2}{x}\frac{3}{x}\\
&+ \frac{x-1}{x}\frac{x-2}{x}\frac{3}{x}\frac{3}{x} \qquad (4.9)
\end{aligned}$$

In all of these the first parasite arrives and destroys down to x eggs. The first line defines the probability that the first parasite is destroyed by the second (which happens with probability $1/x$), the second parasite to arrive is destroyed by the third (again with probability $1/x$), the fourth arrives and destroys a host egg (with probability $\frac{x-1}{x}$) and the fifth destroys a host egg (with probability $\frac{x-2}{x}$). This leaves eggs of the third, fourth and fifth parasite to arrive in the nest. The other lines are the different $\binom{4}{2} = 6$ possibilities.

Equation 4.8 can be simplified to

$$\phi(5,3) = \frac{1}{x^4}\left[\prod_{i=1}^{2}(x-i)\right]\sum_{j_1=1}^{3}\sum_{j_2=j_1}^{3}(j_1 j_2). \tag{4.10}$$

We can now extend this to any value of n_p and x_p, to give

$$\phi(n_p,x_p) = \frac{1}{x^{n_p-1}}\left[\prod_{i=1}^{n_p-x_p}(x-i)\right]\sum_{j_1=1}^{x_p}\sum_{j_2=j_1}^{x_p}\sum_{j_3=j_2}^{x_p}\cdots\sum_{j_{np-xp}=j_{np-xp-1}}^{x_p}(j_1 j_2 j_3\ldots j_{np-xp}). \tag{4.11}$$

The probability that if there are y eggs in the nest in Stage 4 that y_p of them are parasites

With the adult game we have the same stages as before but there is the possibility of more parasites instead of just one and the mean number of parasites that arrive is set at λ_p, so if we assume that y eggs have reached this stage we can work out the value for $\alpha(y,y_p)$.

$$\alpha(y,y_p) = P(Y_p = y_p|Y=y) = \frac{P(Y_p = y_p \cap Y = y)}{P(Y=y)}, \tag{4.12}$$

Where Y is the total eggs that make it to Stage 4 and Y_p is the number of parasites which make it Stage 4. We can work out $P(y)$ as we have with previous games this comes to

$$P(y) = \sum_{n_p=0}^{\infty}\mu_{\lambda p}(n_p)\sum_{x=0}^{n_h}\delta^A_{n_h,x}s_y^x.$$

For the value of $P(y_p \cap y)$ we have the following

$$P(y_p \cap y) = \sum_{n_p=0}^{\infty}\mu_{\lambda p}(n_p)\sum_{x=0}^{n_h}\delta^A_{n_h,x}\sum_{x_p=0}^{max(x,n_p)}\phi(n_p,x_p)s_y^x Y(x,x_p,y_p),$$

where

$$Y(x,x_p,y_p) = \begin{cases} \frac{\binom{x_p}{y_p}\binom{x-x_p}{y-y_p}}{\binom{x}{y}} & if\ x_p > y_p\ and\quad x-x_p > y-y_p, \\ 0 & \text{otherwise.} \end{cases}$$

Just like before we cannot have more parasites or hosts in later stages than were originally laid or that arrived in the nest, so we set $Y(x,x_p,y_p)$ to zero in this case, with a multinomial probability otherwise. So

$$\alpha(y, y_p) = \sum_{x_p=0}^{max(x,n_p)} \phi(n_p, x_p) Y(x, x_p, y_p). \tag{4.13}$$

From equation (4.13) we can create the matrix α

$$\begin{array}{ccccccc} \alpha(0,0) & 0 & 0 & \dots & 0 & 0 \\ \alpha(1,0) & \alpha(1,1) & \dots & 0 & 0 & \\ \vdots & & & & \vdots & \\ \alpha(n_h,0) & \alpha(n_h,1) & \alpha(n_h,2) & \dots & \alpha(n_h,n_h-1) & \alpha(n_h,n_h) \end{array}$$

which only has entries above the leading diagonal as zeros because $y_p \leq y$.

We allow the host to make one of three decisions again as in the previous section.

(a) Proceed and continue to Stage 5

(b) Attempt to eject the parasite egg(s)

(c) Abandon the nest

Rewards are as follows. First, assuming that the host chooses to continue to nurture the nest, we define

$$\Omega(m, m_p, \alpha(m, -)) \tag{4.14}$$

as the outcome for the parasite in the chick game for m total eggs, m_p parasites and parasite probabilities of the m^{th} row of the matrix α defined above. From this we get

$$A(y) = \Omega(y, y_p, \alpha(y, -)). \tag{4.15}$$

Choosing to eject the parasites gives a reward of $B(y)$;

$$B(y) = \sum_{y_p=0}^{max(y,x_p)} \alpha(y, y_p) \left[\sigma\Omega(y - y_p, 0, [0, \dots, 0]) + (1 - \sigma)\Omega(y - y_\sigma, y_p, \alpha(y, -))\right]. \tag{4.16}$$

There is the possibility of an incorrect decision by the host and we must decide how many of its own eggs are ejected, which is denoted by y_σ. As described earlier we choose y_p corresponding to $\max_{y_p>0} \alpha(y, y_p)$, the most likely number of parasites in the nest (note this must be at least one).

Abandoning the nest gives

$$C(y) = -C_A. \tag{4.17}$$

We thus get an overall reward from equations (4.15), (4.16) and (4.17) to the host of

$$\Psi(y) = aA(y) + b(B(y) - C_e) + cC(y). \tag{4.18}$$

Stage 2

Stage 2 becomes the most interesting and therefore the most involved of the stages in the new game. In previous games we only needed to worry about the prospect of one parasite laying and destroying and only with hosts. Then calculating its reward. In this game there are multiple parasites laying. Therefore we have the possibility that a parasite egg gets destroyed by another parasite that lays after it.

We also have the issue that the first parasite chick to hatch gets a different reward from the others. So we must look at how particular parasites can become the first parasite to hatch. We assume that parasites eggs hatch in the same order in which they were laid. Therefore in order to become the first one to hatch two things must happen.

1. All the parasites laid before it, are destroyed by it, by other parasites, nature or the host.
2. It survives.

So we get the following

$$\Pi(n_h, x) = \sum_{n_p=0}^{\infty} \mu_{n_p} \sum_{x_p=1}^{n_p} \frac{1}{n_p} \phi(n_p - 1, x_p - 1)\pi_1, \tag{4.19}$$

where

$$\pi_1 = \sum_{\nu_p=1}^{n_p} \left(\frac{x-1}{x}\right)^{n_p-\nu_p} \sum_{y=0}^{x} s_y^x \frac{y}{x} \pi_2, \tag{4.20}$$

and

$$\pi_2 = \sum_{z_p}^{\nu_p-1} \frac{\binom{\nu_p-1}{z_p}\binom{n_p-\nu_p}{x_p-z_p}}{\binom{n_p}{x_p}} \pi_3, \tag{4.21}$$

and

$$\pi_3 = \sum_{y=0}^{x} s_y^x \left(\frac{z_p}{x-y}\right)^{z_p} (a_y + (1-\sigma)b_y). \tag{4.22}$$

Breaking this down equation (4.19) gives the probability that there are n_p parasites which arrive multiplied by the possibility our focal parasite is the ν_p^{th} to arrive (we set this as uniform for all positions) multiplied by the probability that there are $x_p - 1$ others that will eventually survive until the end. Equation (4.20) shows the probability that it survives both the destruction by the other hosts and nature in Stage 3. Equation (4.21) describes the probability that the $\nu_p - 1$ that arrived before have been destroyed to z_p by the other parasites that arrive in the nest. Equation (4.22) shows the probability that the remaining z_p are all destroyed naturally and the probability the parasite survives Stage 4. These are then multiplied together in equation (4.19) to give the overall probability.

Stage 1

From elements already described we can calculate the host's overall reward, which is given by

$$H(n) = \sum_{y=0}^{n} P(y)\Psi(y). \tag{4.23}$$

Results

We ran simulations of this model using the same cuckoo and cowbird information setting $\lambda_p = 0.06$ as this corresponds to $\beta \approx 0.06$. The results of the game were the same with the cuckoo chick destroying and the cowbird chick leaving the nest as it is. There were also no significant changes in the overall host and parasite reward. The lack of changes in the parasite and the cowbird-type hosts are most likely due to low parasitism rates (the chances of doubling your raising costs being small) but the lack of change in the host rewards in the cuckoo case is because the overall game for the host does not change. It just means its eggs are being ejected by a later arriving cuckoo adult (to replace with one of its own which also later gets ejected) rather than the chick.

4.2.3 Analysing the chick game

In the multiple parasite game we still get an overall expected reward for the parasite as the result of combining those occasions when it is or becomes the first parasite to hatch and those in which it does not. To look into how having more than one parasite in the nest affects the outcomes we must look at the chick game.

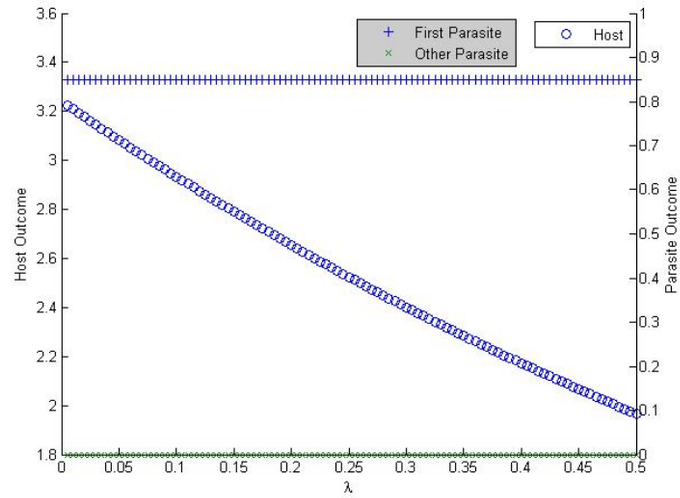


Figure 4.1: The change in parasitism level compared to the outcome for the Cuckoo and Host.

The Parasitism Rate λ

As we can see in Figure 4.1 the host gets a decreased outcome for an increase in the parasitism rate. However, there is no change for the first parasite, as we would expect since this is the destroying cuckoo, and the outcome for the other parasites is zero across the board.

For the cowbird (Figure 4.2) we get a similar pattern to that of the cuckoo, but since it is not a destroyer then the other parasites also get a non-zero reward. The reward for the parasite does not fluctuate because we assumed that its reward is not reduced any more if raised in a nest populated by other parasites compared to being raised with just hosts.

4.2.4 The Full Model

Parasitism Rate λ

Figure 4.3 shows that the increase in parasitism decreases the outcome for both host and cowbird. Note that this is now a curved graph due to the use of a Poisson estimation. The graphs in the previous section showed a linear relation between the parasitism rate β and the rewards (between jumps).

Figure 4.4 shows that the overall trend is a decrease in both host and parasite rewards with three main jumping points. For very low λ, we get the result that the host lays four eggs and raises at the end. In the second subsection at around 0.2 the host only lays three eggs. In the third subsection it lays only two. Both this and the previous section showed similar patterns with regards

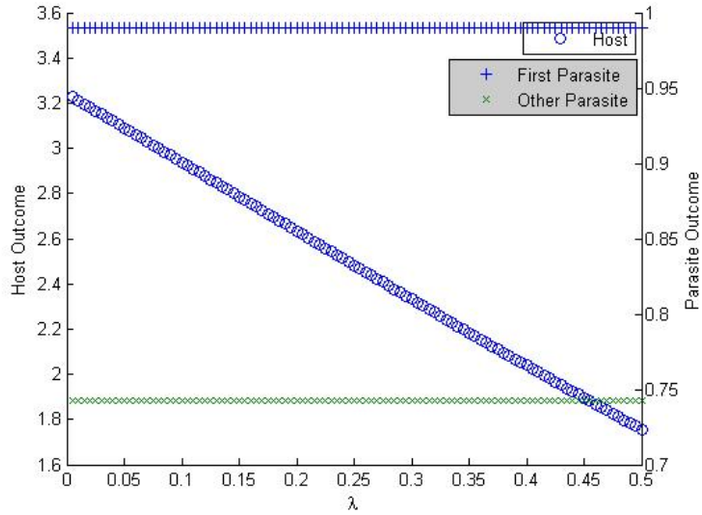


Figure 4.2: The change in parasitism level compared to the outcome for the cowbird and host.

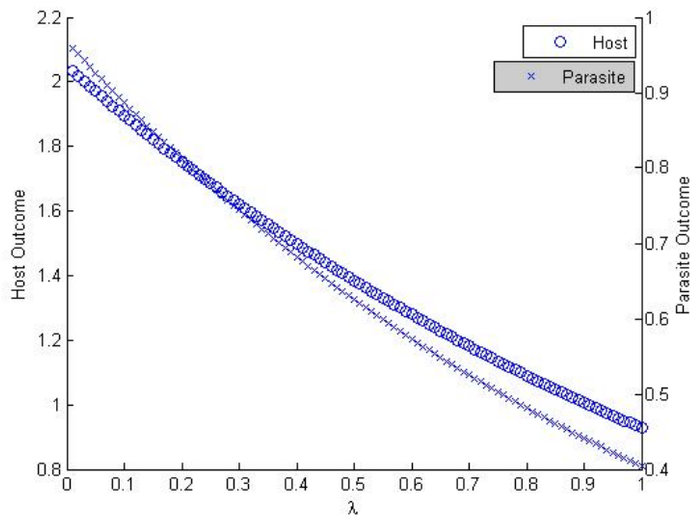


Figure 4.3: Graph showing the effect of increasing parasitism for the cuckoo and hosts.

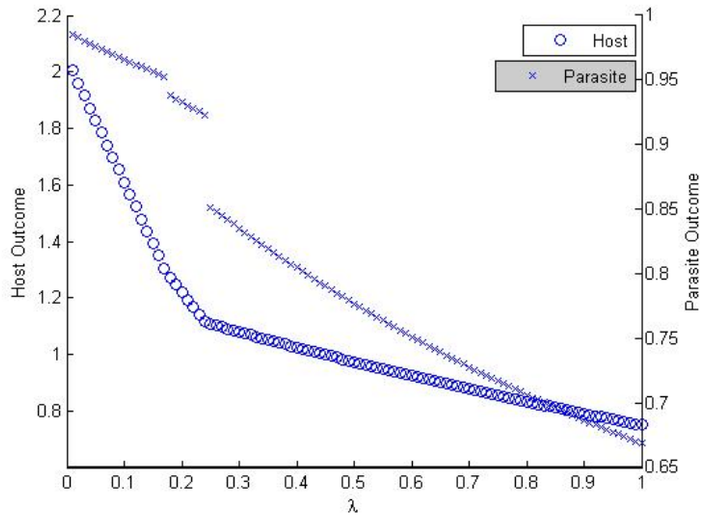


Figure 4.4: Graph showing the effect of increasing parasitism for the cowbird and hosts.

to the decrease in reward when the parasitism rate increases. The major effect of this section is on the parasites. The increased parasitism has a greater negative effect on the parasites than the hosts due to the parasites playing against each other. The host's reward decreases at about the same rate but with a different pattern because of the difference in the way this reward is calculated.

Increasing to very large λ

In the previous examples we have only put λ up to 1 which is quite high but we shall see what happens when the parasitism rate gets very high. Figure 4.5 shows that the rewards for both host and parasite continue to decline as in Figure 4.4 but once λ becomes too high at just over 1.5 the host chooses to only lay one egg. Figure 4.6 shows a similar pattern to Figure 4.5 after the initial changes with λ being less than one from Figure 4.3 and with the host only raising one after λ goes above 1. This is looking at the case where there is a large number of parasites in a particular area and shows that there is a rate at which the host reward will trend in a similar way to that of the parasites before dropping to zero at very high λ.

4.2.5 Discussion

Improving the previous games to include the possibility of more than one parasite arriving at the nest is a necessary addition, because even if the chances of this occurring are small it still could potentially happen, and it becomes important for higher parasitism rates. Changes in the value of

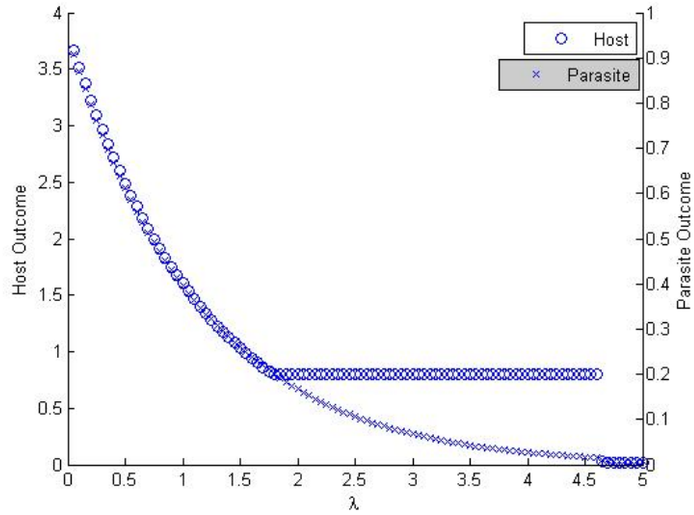


Figure 4.5: Reward for host and cuckoo for large values of λ.

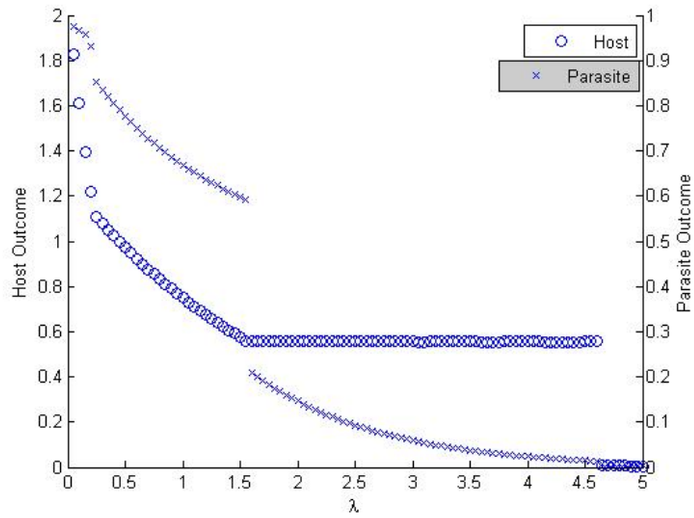


Figure 4.6: Reward for host and cowbird for large values of λ.

our λ show differences in reward for both parasites and hosts. As we would expect, the destroyer type Cuckoos are worse off if parasitism increases since they wish to be raised alone, however this leads to them playing against each other. This also occurs for the cowbird-type parasites, but since the chicks do not display any destruction tendencies the chances of all the parasite birds being hatched and raised is higher. Fraga [12] has shown situations where up to 19 parasites can visit a nest. Johnsgard [20] also mentions these possibilities in his book entitled *The avian brood parasites: deception at the nest.*

We consider real examples of brood parasitism and find as many of our parameters for these situations as possible, with other parameters and functional forms chosen to be as realistic as possible. We find that the solutions from our game match with those from reality reasonably well. In particular changes in just a single parameter within the reward function for the host moves the population from one of the classic behaviours of destruction of all the host young by the parasite chick, associated with cuckoos, to the other classic behaviour of no such destruction associated with the cowbirds. Studying the behaviour of the solution as this parameter changes shows intermediate solutions which occur for small parameter ranges only. Thus perhaps real behaviour can be explained in terms of individuals making optimal choices in games under certain constraints which are the same for most brood parasitism situations, and do not need to rely on evolutionary lag and/or size restriction arguments.

In this sequential game, there are different places where the game can stop, for example abandonment of the nest by the host after the parasite has laid or at the moment of hatching. If modeled in real time there would not be arbitrary costs associated with staying in or leaving the game at a given time; the cost would be related to how much time could not be used in future activities because it is being used to participate in this game (e.g. the opportunity to start a new nest for the host). We can develop a model where there are multiple hosts and have interactions taking place in real time, so that individuals play the game as above but leave the game at different times. The reward for leaving at a given time for a host will depend upon various ecological parameters such as the time remaining to the end of the breeding season, but also the number of laying hosts per parasite in the population at that time. When should the hosts lay in such a population? It seems likely that they should lay together, not just because it is the best time for reasons associated with ecological parameters, but because it will dilute the effect of the parasites. These and other types of effects are investigated in the next chapter.

Chapter 5

Time-Dependent Multiple-Host Model

5.1 Introduction and Notation

We now wish to extend the model so that we have interactions between many hosts and many different parasites over a breeding season. We have previously introduced the notion of abandoning a nest and explained that this relates to the idea of re-nesting later on in a breeding season. But what if it is not possible to do so or there are some outside elements which could affect relaying? We have also looked at the possibility of the parasites having competition between themselves so it would be prudent to see how the interactions between the individual hosts affect the interactions of hosts and parasites. We shall look at different aspects of laying behaviour to see how this affects the individual interactions we have previously modelled.

We shall break down the model into discrete time periods where the interaction would happen. We shall further break the single interaction model down into three distinct periods,

a) Laying of the egg, a parasite arriving with a probability and some natural destruction (S1 - S4)

b) The host choosing to nurture the eggs until hatching (S4 - S7)

c) The host choosing to raise the chicks (S7-).

We say that each of these last for a given time period τ_a, τ_b and τ_c ($\tau_j \in \mathbb{N}$), where the host has an opportunity to opt out after point a and b and the natural end is after c. Thus the full length

of the game is

$$\tau = \tau_a + \tau_b + \tau_c \tag{5.1}$$

We say that the entire length of the breeding period is T. We break this down into $m = T - \tau$ starting points labeled $t \in \{1, 2, \ldots, m\}$. This is because the host must still have the time left in the breeding season to raise the brood.

Points to note

- The earliest starting period is $t = 1$.
- The latest point at which the host can start to raise a brood is $t = T - \tau = m$
- If the host decides to abandon the nest it may start another nest, as long as it starts the first nest by at least $t = m - (\tau_a + \tau_b)$
- The host may also start a new nest if it starts at $t = m - (\tau_a)$ as long as it abandons after a.
- In order to simplify the model we assume that the host can only raise one clutch once during the season. Even though this does not match up exactly with that in nature (for all species of hosts). The key elements of this interaction should still hold true.

We suppose that there is a definite number of hosts (n_h) that are going to breed throughout the season. The number of hosts nesting at a time point t is labeled n_{h_t}.

It is only necessary to look at n_h different nesting sites since it is only possible for n_h hosts to nest in a given time point and we also assume there are actually enough nesting sites for the hosts to find a nest, we label each of the nests θ_i with $i \in (1, n_h)$. We also assume a ν_i which represents the *value* of nesting site θ_i. This parameter determines the likelihood a parasite will discover the nest. This factor could be based upon line of sight for the parasite or many other aspects. We set the nesting sites up so that

$$\nu_1 \leq \nu_2 \leq \cdots \leq \nu_{n_h},$$

meaning the smaller the nesting site parameter the better the nest is. If there is only one host that lays at a given time point it automatically gets nesting site θ_1. If there are two we suppose it has

Table 5.1: Generalised Table of Nesting Site Usage over the breeding season

Site	$t=1$	2		τ	$\tau+1$	$\tau+2$	
θ_1	h_1	h_1	...	h_1	$h_{\tau+1}$	$h_{\tau+1}$	...
θ_2	h_1	h_1	...	h_1	$h_{\tau+1}$	$h_{\tau+1}$	...
...							
$\theta_{n_{h1}}$	h_1	h_1	...	h_1	?	?	...
$\theta_{n_{h1}+1}$	$-$	h_2	...	h_2	h_2	?	...
...							
θ_{n_h}	$-$	$-$	...	$-$	?	$-$	...

a fifty percent chance of getting either nest θ_1 or θ_2 and so on. This will come into play when we work out the choice for how many of the hosts will lay at a given time point.

We must work out how these sites are populated. Suppose we have a decision by the n_h hosts as,

$$[n_{h_1}, n_{h_2}, \ldots, n_{h_{(t-\tau)}}, 0, \ldots, 0]$$

where n_{h_t} is the number of hosts laying at time point t.

We get the n_h nesting sites populated as shown in Table 5.1, where h_i represents a host who decides to lay in time $t = i$. The host who lays in time point $t = 2$ has to lay in nest $\theta_{n_{h1}+1}$ because the previous nests are being used by those hosts who chose to lay in the first time point. Those who choose to lay at time point $t = \tau + 1$ can then begin to lay in nest θ_1 because the hosts will have vacated those nesting sites. We fill some of the points with question marks because we don't know if the nests will be occupied at that given time point. For example the question mark in the square $\theta_{n_{h1}}, \tau + 1$ could be filled with $h_{\tau+1}$ if there are more hosts laying in time point $\tau + 1$ than time point 1. If there are more hosts laying at time point one, nest $\theta_{n_{h1}}$ will be unused at time point $\tau + 1$. The different permutations of nesting sites used make it difficult to generalise these situations.

We look at an example with $n_h = 5, T = 6$ and $\tau_a = \tau_b = \tau_c = 1$. If we assume a decision of $[1, 2, 1, 1]$, we get the site usage as explained in Table 5.2.

Thus the first host to lay gets the best nesting site. The hosts which lay in the second and third time points must take the next available sites. Since the first host has finished nesting at $t = 3$, the host which begins at $t = 4$ now gets the best nesting site.

If one of the hosts chooses to abandon its nest then we must re-arrange this pattern. For

Table 5.2: Nesting site usage for 5 hosts laying in 6 time points with the decision $[1, 2, 1, 1]$

Site/Time	1	2	3	4	5	6
1	1	1	1	4	4	4
2	0	2	2	2	0	0
3	0	2	2	2	0	0
4	0	0	3	3	3	0
5	0	0	0	0	0	0

Table 5.3: Nesting site usage for 5 hosts laying in 6 time points with the decision $[0, 3, 1, 1]$

Site/Time	1	2	3	4	5	6
1	0	2	2	2	0	0
2	0	2	2	2	0	0
3	0	2	2	2	0	0
4	0	0	3	3	3	0
5	0	0	0	4	4	4

instance if the host that originally laid their nest in time point one chooses to abandon and then immediately re-lay we must look at the change in overall decision. In this instance it becomes $[0, 3, 1, 1]$, which gives the nesting pattern in Table 5.3. This decision does not affect the pattern of the ones originally laying in the second and third time points, but the reward will change because the hosts have a choice of three nests rather than just two. The host laying in the fourth time point must now settle for the fifth best nesting site.

5.2 Nesting site values

We also assign a value to the difficulty for the parasite to find and therefore lay an egg in the nest. We say that each of the i nests have a value ν_i. This is a number which can be used to calculate the parasitism rate λ. If we have the number of hosts laying at a time point t (n_{h_t}) and the number of parasites at time point t (n_p^t), then

$$\lambda_{i,t} = \frac{\nu_i n_p^t}{n_{h_t}}. \tag{5.2}$$

This means that the parasitism rate increases with the number of parasites at a given time point or a higher nesting site value and decreases with the number of hosts laying at a given time point.

The host has the choice to start at any time point $t = 1$ up to $t = T - \tau$. For simplicity of

calculation we assume it will only lay one clutch in a season, so therefore chooses t to maximize its outcome. We denote the expected total outcome for the host at each time period by $\Omega_{t,i}(A)$, where

$$A = [R_H(x,t,i), C_R(x,t,i), C_L(x,t,i), C_a(t,i), C_b(t,i), \lambda_{i,t}] \tag{5.3}$$

where $R_H(x,t,i)$ is the reward to the host for raising x eggs at time point t in nesting site i, $C_R(x,t,i)$ is the cost to raising x eggs at time point t in nesting site i, $C_L(x,t,i)$ is the cost to laying x eggs at time point t in nesting site i, $C_a(t,i)$ is the outcome it will get if it abandons at Stage 3, $C_b(t,i)$ is the outcome it will get if it abandons at Stage 7 and $\lambda_{i,t}$ is as described in equation (5.2).

5.3 Calculating the host decisions

There are some assumptions which come out of this model.

- $\sum_{t=1}^{m} n_{h_t} = n_h$

  Since we have assumed the host can raise a brood exactly once (even though it can lay again after abandoning) in a breeding season, the total number of hosts breeding is the summation of all those breeding at a given time point.

- $\max(n_h^i) = n_h$

  The number of hosts laying at a given time point can not exceed the total number of hosts in the game.

This gives us a definitive number of different combinations of what the hosts as a group do. i.e. how many of them lay at each of the given time points. We can represent these decisions in the following matrix.

$$C = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & n_h \\ 0 & 0 & \dots & 0 & 1 & n_h - 1 \\ 0 & 0 & \dots & 1 & 0 & n_h - 1 \\ \dots & & & & & \\ \dots & & & & & \\ n_h - 1 & 1 & \dots & 0 & 0 & 0 \\ n_h & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

This matrix shows each possible decision the host can make. The first row is the decision for all n_h hosts to lay at the final time point. The second is that all, apart from one, lay in the last time point and the other lays in the second to last time point. All combinations are considered following this pattern until the final row which is the decision for all to lay at the first time point.

We then look for which of the solutions are stable. (i.e. cannot be invaded by an individual host). For example if we have two time points ($T-\tau = 2$) and two hosts. If both possible nests have the same nesting site value parameter it will always be better for the hosts to lay together, which means that $[2, 0]$ and $[0, 2]$ will be stable solutions (because they cannot be invaded successfully by a host that chooses to lay on its own).

For interest, we can work out the value of $\Omega_{t,\theta}(A)$ for each line of the matrix. From this we can deduce which line of the matrix produces the best outcome for the hosts as a population and see if this outcome is stable, by which we mean that if any host chooses to make a different choice it must always end up being worse off. There may be many stable solutions but in general there will be an optimal choice. If the costs and rewards are uniform over the breeding season then $[2, 0]$ should be an optimal solution because the hosts have a chance to re-lay if they choose to abandon after *Stage 4*.

5.4 Calculating Expected Rewards for the Host

We must also consider what happens after abandonment. We assume that a host which abandons can re-lay its nest from any point later on in the season, but how does this affect the other hosts which lay later on? The corresponding line of the matrix C will differ, for example if we have a line

$$[1, 0, 0, 2]$$

and the host in the first time point chooses to abandon (with $\tau_a = 1$), we get three different possible vector outcomes

$$[0, 1, 0, 2], [0, 0, 1, 2], [0, 0, 0, 3].$$

Thus we must work out the matrix C in the following way. First $[0, 0, \ldots, 0, n_h]$ is worked out because there can be no re-nesting in this decision. Then we work out $[0, 0, \ldots, 1, n_h - 1]$ because the only abandonment scenario for this one is $[0, 0, \ldots, 0, n_h]$, and we continue this pattern (where the only abandoning scenarios are the ones we have already calculated) until the final row which will be $[n_h, 0, \ldots, 0, 0]$. From this we can then choose which of the abandonment scenarios is best for the host. The expected outcome from choosing to re-lay is taken from this previously calculated scenario and is fed into the single interaction model as the value for C_a and C_b (the cost/reward for abandoning at Stage 4 and Stage 7 respectively). Note that the first line of matrix C is always calculated with $C_a = C_b = 0$ because there is no chance of re-laying.

We can now calculate an expected reward ($\Omega_t(A)$) for each of the birds in each of the nesting patterns where A comes from Equation (5.3). If we assume a nesting pattern

$$[n_h^1, n_h^2, \ldots, n_h^t, \ldots, n_h^{T-3}],$$

$C_a(t)$ is worked out as the maximum of the following lines in C

$$\left\{[n_h^1, n_h^2, \ldots, n_h^t - 1, \ldots n_h^{t+1} + 1 \ldots, n_h^{T-3}], \ldots, [n_h^1, n_h^2, \ldots, n_h^t - 1, \ldots, n_h^{T-3} + 1]\right\},$$

these being the abandonment scenarios for a host laying at time point t and choosing to abandon at Stage 3. A similar result can be found for the value of $C_b(t)$

5.4.1 Extracting Abandonment Probabilities from the chick game

Going back to the multiple model in Equation (4.5) the overall reward to the host if it raises x given m made it to Stage 4 was

$$r(x, m) = \sum_{m_p=0}^{m} \alpha_{m_p} \sum_{x=1}^{m-1} \delta^C_{m-1,x-1} \sum_{x_h=0}^{x} M(m_h, x_h) \sum_{y=0}^{x} t_y^x \sum_{y_h=0}^{y} R(x_h, y_h)$$

where

$$M(m_h, x_h) = \begin{cases} \frac{\binom{m_h}{x_h}\binom{m_p-1}{x-x_h-1}}{\binom{m-1}{x-1}} & if\ m_h \geq x_h, \quad and\, m - m_h \geq x - x_h, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$R(x_h, y_h) = \begin{cases} \frac{\binom{x_h}{y_h}\binom{x-x_h}{y-y_h}}{\binom{x}{y}}(R_H(y_h, t, i) - C_R(y_h + (y - y_h)\xi, t, i)) & if\ x_h \geq y_h \quad and\, x - x_h \geq y - y_h, \\ 0 & \text{otherwise.} \end{cases}$$

This is then compared to $C_b(t, i)$ to see if it is viable to continue nurturing with $\rho_x = 1$ if it does and $\rho_x = 0$ if it does not. We can get the probability of m being destroyed down to x as

$$p(x, m) = \sum_{m_p=0}^{m} \alpha_{m_p} \sum_{x=1}^{m-1} \delta^C_{m-1,x-1} \sum_{x_h=0}^{x} M(m_h, x_h) \sum_{y=0}^{x} t_y^x \sum_{y_h=0}^{y} P(x_h, y_h) \tag{5.4}$$

where

$$P(x_h, y_h) = \begin{cases} \frac{\binom{x_h}{y_h}\binom{x-x_h}{y-y_h}}{\binom{x}{y}} & if\ x_h \geq y_h, \quad and\, x - x_h \geq y - y_h, \\ 0 & \text{otherwise.} \end{cases}$$

and the function $M(m_h, x_h)$ is as before. Thus the probability of abandoning in the chick game is

$$A_c(m) = \sum_{x=0}^{m} (1 - \rho_x) P(x, m). \tag{5.5}$$

5.4.2 Extracting Information from the adult game

We can find the probability that there are y eggs in Stage 4 as

$$Q(n_h, y) = \lambda_{i,t}(0) \sum_{y=0}^{n} s_y^n + (1 - \lambda_{i,t}(0)) \sum_{x=0}^{n} \delta_{n,x} \sum_{y=0}^{x} s_y^x. \tag{5.6}$$

From equation (5.6) we can calculate the probability of abandonment as an adult as

$$\sum_{y=0}^{n_h} Q(n_h, y) \left(a_y A_c(y) + b_y A_c(y_b) + c_y\right), \tag{5.7}$$

where a_y is the choice to leave the nest alone, b_y is the decision to eject some eggs (with y_b being the amount ejected down to) and c_y is the decision to abandon the nest. All of these have a subscript y because we have different decisions that the hosts makes dependent upon the number of eggs in the nest at that stage.

We look at a very simplified example with $n_h = 2$ and two different starting time points ($\tau_a = \tau_b = \tau_c = 1$). This means we can only have three different choices and get the matrix C to be

$$C = \begin{bmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}.$$

We set the values $\nu_1 = 1/100$ and $\nu_2 = 5/100$, with $n_p = 6$ in both the two time points and we use the variables for the Cowbird game previously mentioned. We get the following expected outcomes.

$$\begin{bmatrix} 0 & 2.7045 \\ 2.7975 & 2.1665 \\ 2.7149 & 0 \end{bmatrix}.$$

Each of the values represent the expected outcome for each individual host should it choose that nesting pattern. The first row represents both hosts laying at the second time point and receiving 2.7045. The second is one laying at the first time point and getting 2.7975 and one laying at the second time point and getting 2.1665. The third line is both laying at the first time point and each getting 2.7149. This means that $[1, 1]$ is not stable because it can be invaded by both of the other solutions.

Occasionally some of these will throw up solutions that don't make sense. For example, if we have a situation where we have two hosts and only one viable nest, there is time for the hosts to both use this nesting site. However, because the probability that both will be laying together affects the overall outcome, you will get solutions where both will lay at the beginning giving a 50% chance of using both nests with the host that doesn't get the *good* nest immediately abandoning and using the *good* nest later when you would expect it to. In a sense this is not what is truly happening with the game and it can affect which is the best outcome to choose and also which abandonment scenario is best to choose. In reality these will be unstable solutions and dominated by other solutions. For this reason we simply remove the situation where the outcome of A_a or A_c is above 0.99.

Good Nests	Best Choice
1	[1,0,0,3]
2	[2,0,0,2]
3	[3,0,0,1]
4	[4,0,0,0]

Table 5.4: Best nesting pattern choice for the hosts for different numbers of *Good* nests, with four hosts and four possible starting points

5.5 Examples

5.5.1 *Good* and *Bad* Nests

We initially set the game up so there are two distinct types of nesting sites, *Good* nesting sites and *Bad* nesting sites, what we are going to focus on here is the change in host strategy dependent upon the number of these *Good* nests compared to the number of *Bad* nests.

We consider a situation where we have four hosts and four possible starting points. We set A (from 5.3) as the cuckoo parameters from the Single Interaction model for every time point and site. We can set the number of *Good* nests as anything from one to four (any more than four is unnecessary). We set the value of the *Good* nests as $\nu_i = 0.01$, with the six parasites giving us $\lambda = 0.06$ if it lays alone, $\lambda = 0.03$ if there are two hosts laying at the same time and so on. We similarly set $\nu_i = 1$ for the bad nests. Again we assume $\tau_a = \tau_b = \tau_c = 1$.

We get the outcomes shown in Table 5.4, the total reward for each can be seen in Figure 5.1 and follows the trend we would expect in that the more *Good* nests the better an overall reward for the host population. Note that the reward for two and three is not much different because what is gained by the three hosts raising together is partly lost by the one host raising alone. The first decision is $[1, 0, 0, 3]$ rather than $[3, 0, 0, 1]$ because all the hosts will attempt to lay in the first nest in the first time point and rather than take one of the *Bad* nests the three remaining compete for the *Good* nest once it becomes occupied again at time point 4.

If we increase the number of hosts to eight we get a similar pattern and increase in total outcome as shown in Table 5.5 and Figure 5.2 show. Note again that the increase in outcome is less when you increase the nests from four to 7 for the same reasons as in the previous case, in that the increase in more laying together is counterbalanced by others laying with less.

Now we increase the number of time points to ten and we get the outcome as shown in Table 5.6. This is very similar to the results with only four time points except for the situation where there is only one good nest. As expected the hosts spread themselves out amongst the nests. As you can

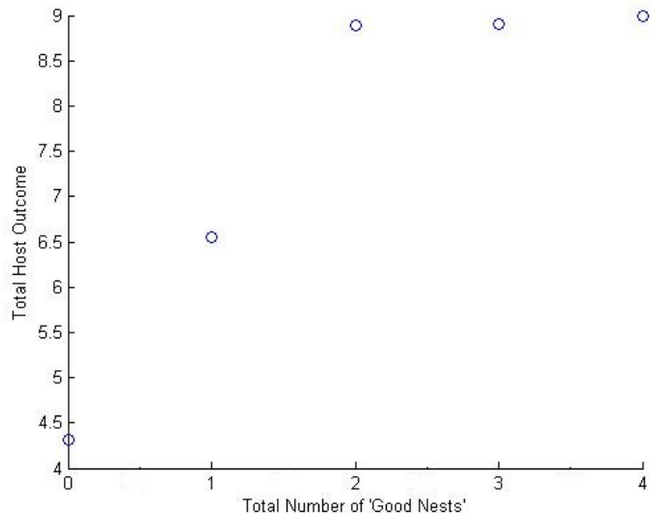


Figure 5.1: The Total Outcome for Hosts with increasing number of good nests with four hosts and four possible time points

Good Nests	Best Choice
1	[1,0,0,7]
2	[2,0,0,6]
3	[3,0,0,5]
4	[4,0,0,4]
5	[5,0,0,3]
6	[6,0,0,2]
7	[7,0,0,1]
8	[8,0,0,0]

Table 5.5: Best nesting pattern choice for the hosts for different numbers of *Good* nests, with eight hosts and four possible starting points

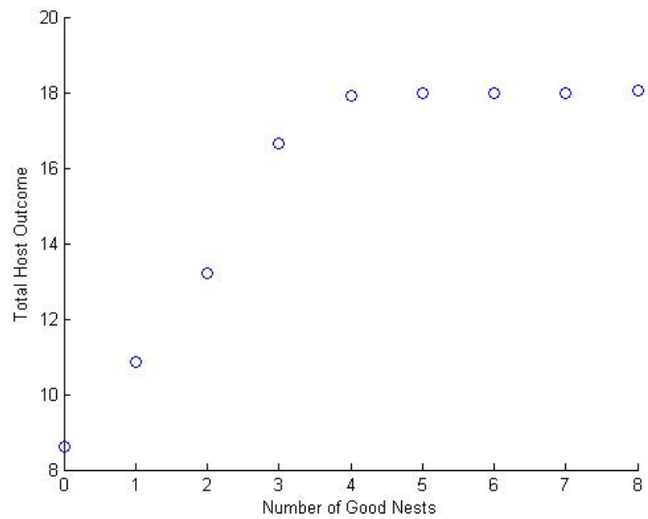


Figure 5.2: The Total Outcome for hosts with increasing number of good nests with eight hosts and four starting time points

Good Nests	Best Choice
1	[1,0,0,1,0,0,1,0,0,1]
2	[2,0,0,2,0,0,0,0,0,0]
3	[3,0,0,1,0,0,0,0,0,0]
4	[4,0,0,0,0,0,0,0,0,0]

Table 5.6: Best nesting pattern choice for the hosts for different numbers of *Good* nests, with four hosts and ten possible starting points

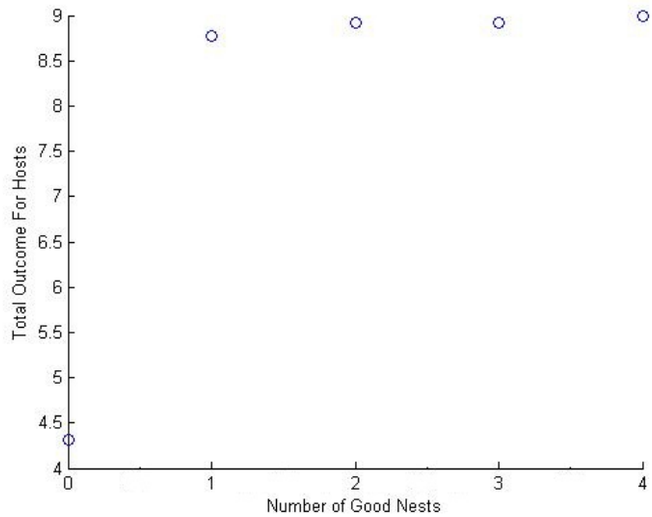


Figure 5.3: The total outcome for Hosts with increasing number good nests, with four hosts and ten possible starting points.

see in Figure 5.3 the increase occurs the same as with the example of four time points (Figure 5.1) with the only significant exception being the one with only one *Good* nesting site. We can note that $[2, 0, 0, 2, 0, 0, 0, 0, 0, 0]$ yields the greatest overall outcome for the hosts. However it is not the only stable solution, in fact $[0, 2, 0, 0, 2, 0, 0, 0, 0, 0], [0, 0, 2, 0, 0, 2, 0, 0, 0, 0], \ldots, [0, 0, 0, 0, 0, 0, 2, 0, 0, 2]$ are all stable solutions in that a host cannot choose to go away from this and yield a larger result. $[2, 0, 0, 2, 0, 0, 0, 0, 0, 0]$ is the choice the hosts make because it has the largest scope for abandonment and relaying of nests.

This shows that the increase in number of nests has a diminishing effect for small number of hosts. But what about larger numbers of hosts. Figure 5.4 shows the effect of increasing *Good* nests on twenty hosts in four possible starting points. As you can see the effect of going from only one or two good nests upwards sharply increases the host reward. Beyond this the increase slows but still increases significantly because all the hosts can now get a *Good* nest.

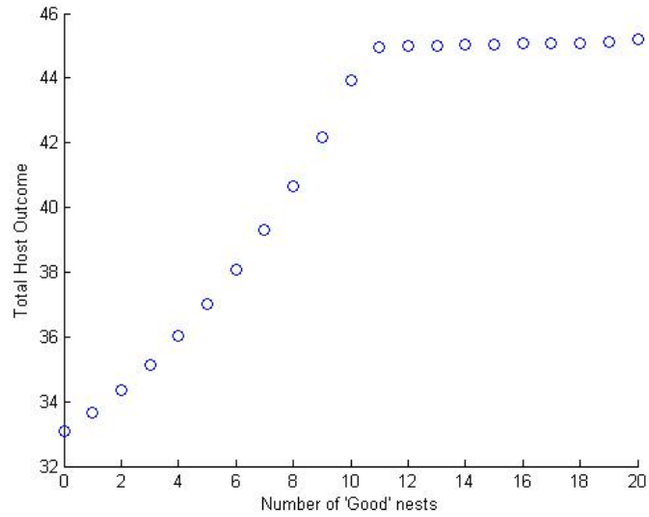


Figure 5.4: The total outcome for hosts with increasing number of good nests with twenty hosts and four starting points

5.5.2 The Number of Hosts

How the number of hosts affects the reward to those hosts is dependent on how many good nests there are. If there are infinitely many good nests then we get the outcome as shown in Figure 5.5, where we get an increased average outcome as we would expect.

We now look at the effect of the number of hosts when we don't just have all good nests. Figure 5.6 shows ten different graphs running from one good nest to ten.

When we have only a few good nests but a lot of hosts we get a decreasing curve in the host outcome. This is because the chance of getting a good nest is less as the number of hosts goes up. As the number of *Good* nests increases we get increasing curves which seem to *jump* down at certain points. Note that this occurs for a number of hosts larger than the number of *Good* nests and is because there will be some drop because they do not get equal benefit of sharing the nests and all having *Good* nests. This decrease continues along until the final graph which is the same as Figure 5.5 because we now have ten nests.

We can extend this to up to twenty hosts. The graphs in Figure 5.7 show the average outcome for up to twenty hosts for different numbers of *Good* nests but this time with only two possible time points (this was done to decrease computation time but does not affect the result). These graphs show that the host reward increases with the number of other hosts until the number of *Good* nests is reached and then there is a sharp decline in host reward. This does continue to increase again

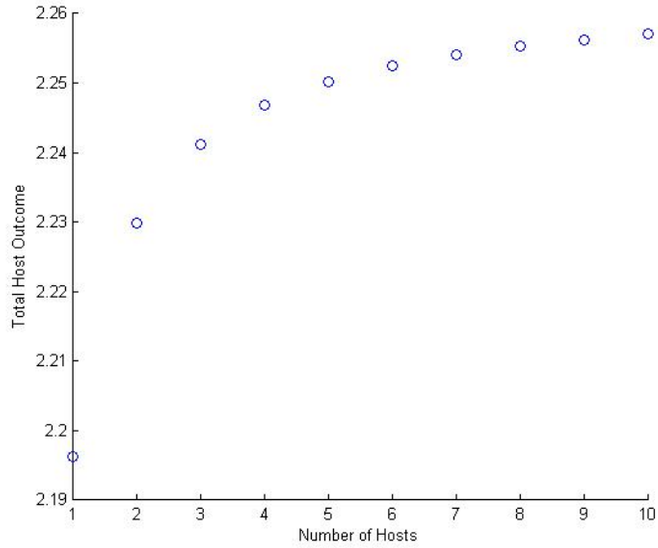


Figure 5.5: Outcome to the host for increasing number of hosts laying in the season

as having very large numbers of hosts takes away the decrease in award of having *Bad* nests. This does not however get to the point where it is better than all the hosts being able to lay in a good nest.

5.5.3 The Number of Parasites

Looking at Figure 5.8 we see how the number of parasites affects the overall host outcome. We set the time points as well as the number of hosts to four. Apart from the occasion where there is only one good nest, the increase in the number of parasites gives a linear decrease in host reward because there is a linear relationship between the number of parasites and the parasitism value λ. In the case where there is only one *Good* nest in the first graph the effect of increasing the number of parasites has a greater initial decrease as we go from no parasites to two but once we get to three parasites or more the decrease is linear like the other graphs. This is because having fewer parasites is more beneficial if there is only one *Good* nest.

5.5.4 Adding in the possibility of *Average* nests

Previously we have looked at the possibility of just *Good* or *Bad* nests. Instead of the binary choice of good and bad nests we can add the possibility of some *Average* nests and seeing what effect it will have.

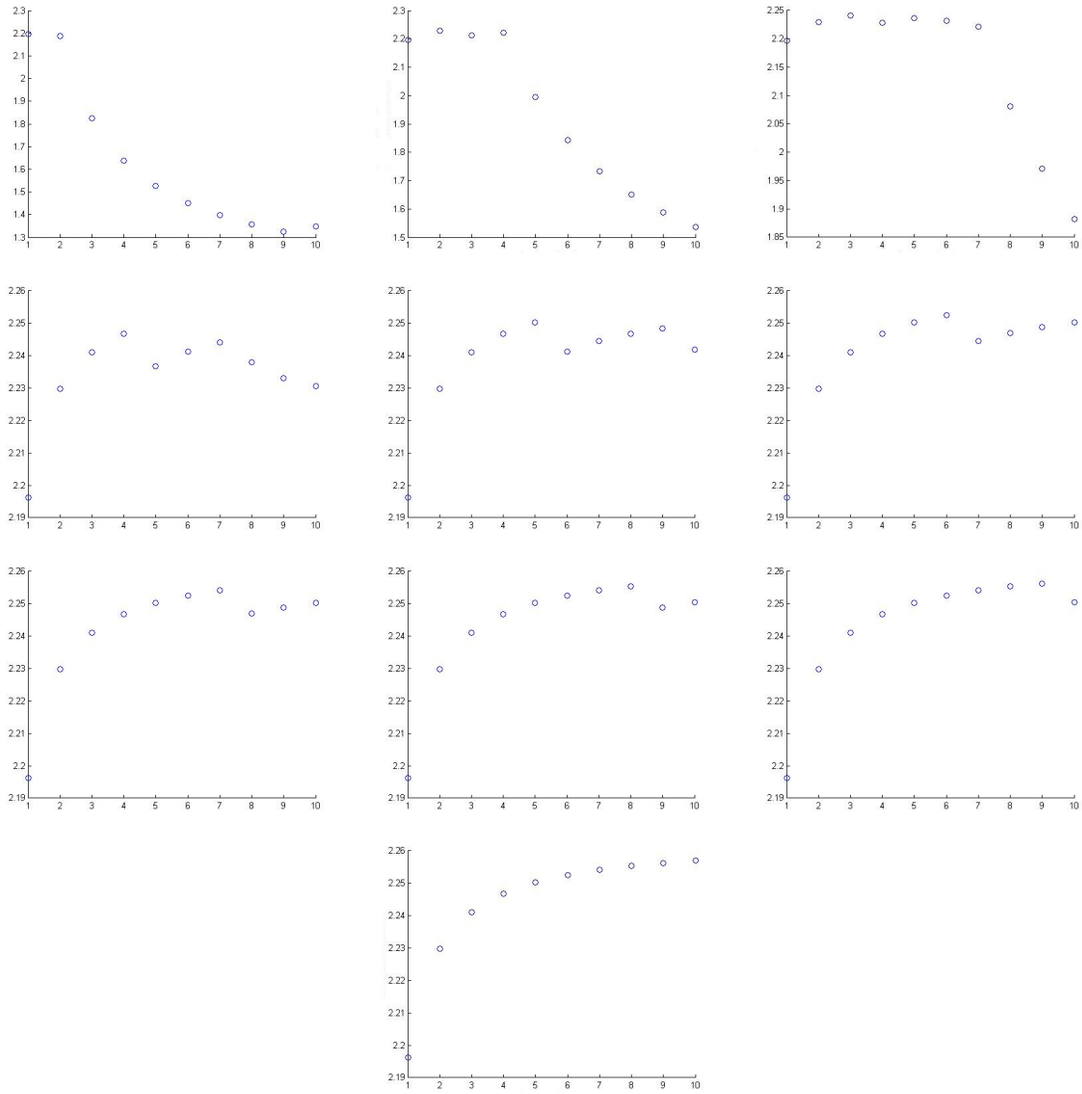


Figure 5.6: The Average Outcome for Hosts for increasing number of *Good* nests and Increasing number of hosts. Reading from left to right the first graph is that where there is one good nest. The second has two *Good* nests and so on until the last which has ten *Good* nests. On each of the graphs the vertical axis is the host outcome and the horizontal axis is the number of hosts

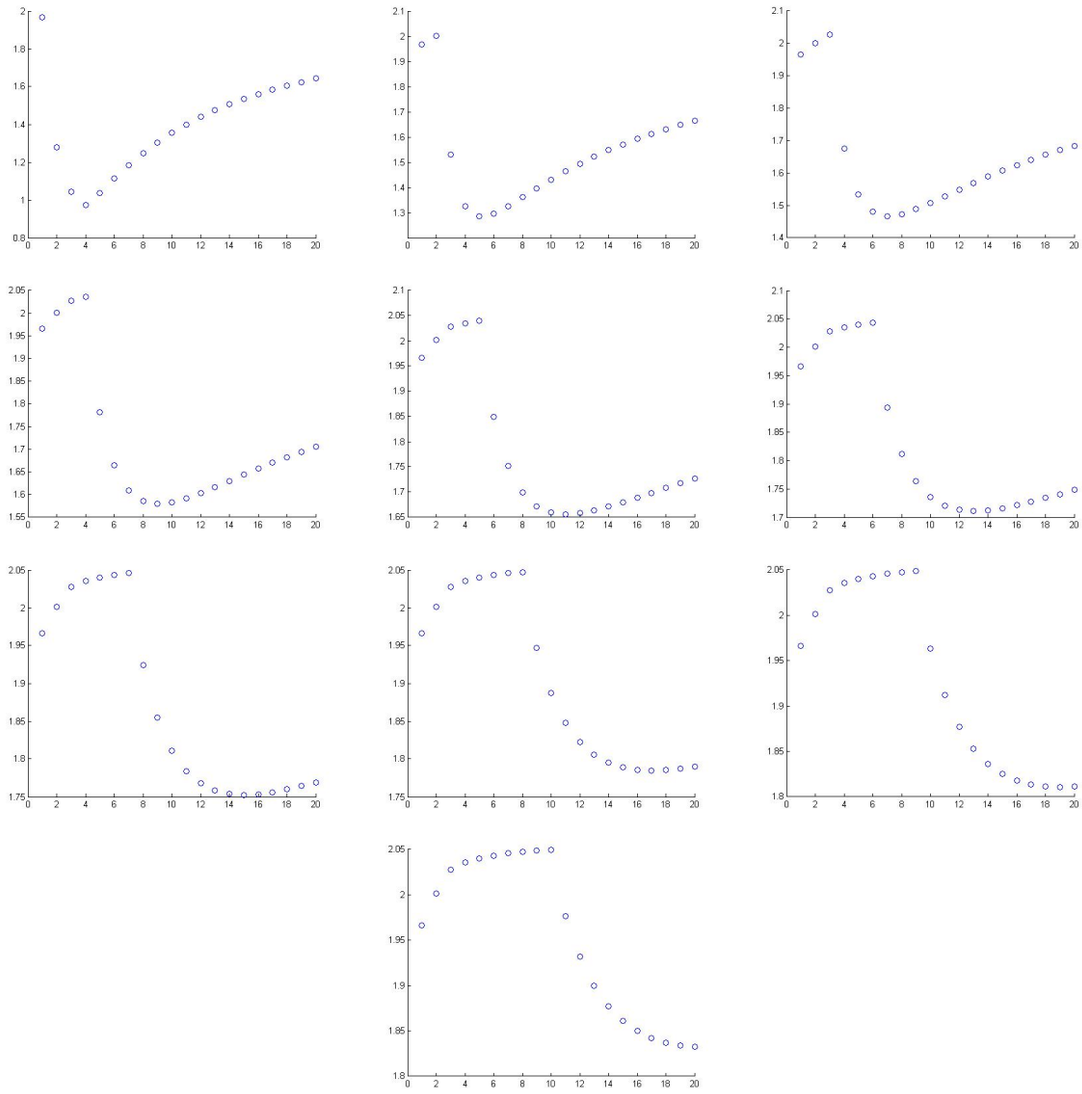


Figure 5.7: The Average Outcome for up to twenty hosts for increasing number of *Good* nests and increasing numbers of hosts with two starting time points. Reading from left to right the first graph is where there is one good nest. The second has two *Good* nests and so on until the last which has ten *Good* nests. On each of the graphs the vertical axis is the host outcome and the horizontal axis is the number of hosts.

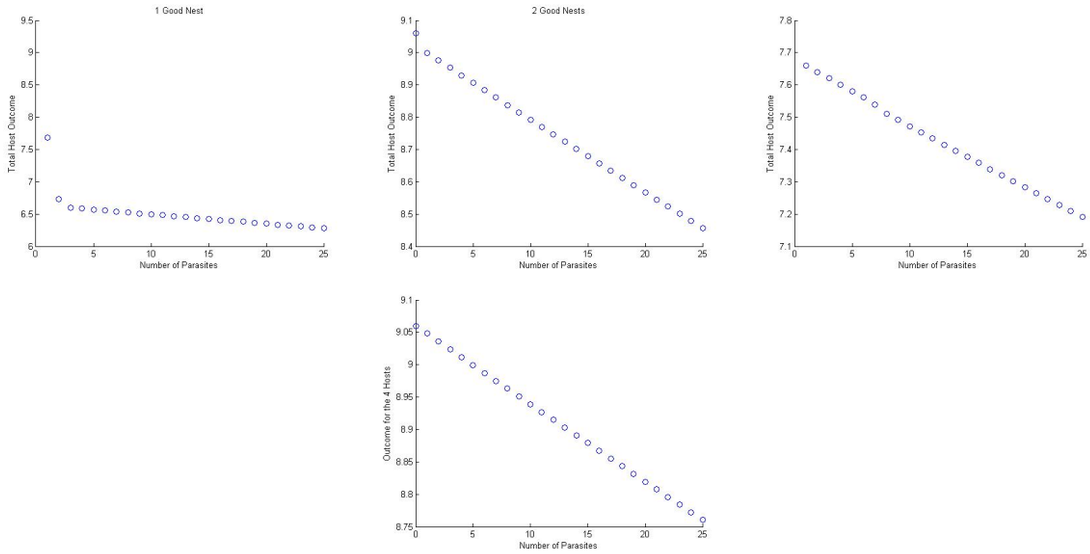


Figure 5.8: The Total for Four Hosts for Increasing Number of Parasites and *Good* Nests. The first graph has 'one' good nest, the second has 'two' and so on.

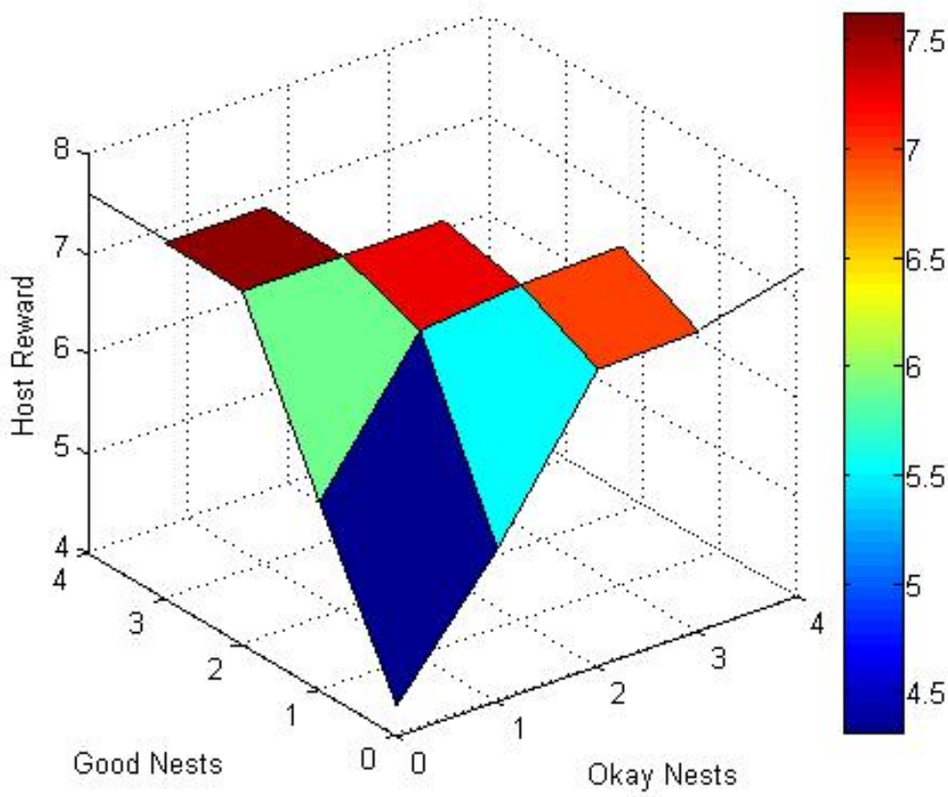


Figure 5.9: Three Dimensional diagram of Good and Average nests versus the Total Outcome of Four Hosts. The colour of the square represents the point closest to the $(0, 0, 0)$ point.

Figure 5.9 shows the effect of the good and average nests. Along the *Average* and *Good* axes we can see a similar pattern to that of Figure 5.1 but with the overall reward of the *Good* nests being higher than the *average* nests as we would expect. This pattern continues with the other combinations. The host treats an *Average* nest as it did *Bad* ones in the previous examples, if there are *Good* ones available and treats the *Average* nests as *Good* ones if there are only *Average* and *Bad* nests available. As we can see the overall reward for having more *Average* nests is less as the colours are lower on the scale.

5.5.5 Different Costs During the Season

Another aspect of this game we consider is seeing if there is an effect of the cost to raise a brood differing during a breeding season. At the moment the cost of raising is a function on the number of eggs (x) in the nest at that time is given by

$$C_R(x) = \nu e^{x/2}, \tag{5.8}$$

where ν is a scalar, which in previous examples has been around 0.25. We can now look to alter this so it has a different outcome depending on the choice of starting time point. This makes ν a function of t. Thus we make C_R a function of both x and t and choose

$$C_R(x,t) = \nu(t) e^{x/2}, \tag{5.9}$$

so now we must look at the different function for $\nu(t)$.

Laying is better in the middle of the season

First of all we set up the function $\nu(t)$ such that the lowest cost of raising is in the middle of the breeding season. We set

$$\nu(t) = \left(\frac{t - \frac{m+1}{2}}{m+1} \right)^2 + 0.1, \tag{5.10}$$

where m is the number of time points. This gives us the curve for $\nu(x)$ in Figure 5.10. We start with an example with two hosts and five starting time points. We can show that the stable solutions are those shown in Table 5.7. In this case the optimal solution is not $[2,0,0,0,0]$ as it would have been with a scalar $\nu(t)$. The lowest cost of raising is to start in the middle of the game and we get

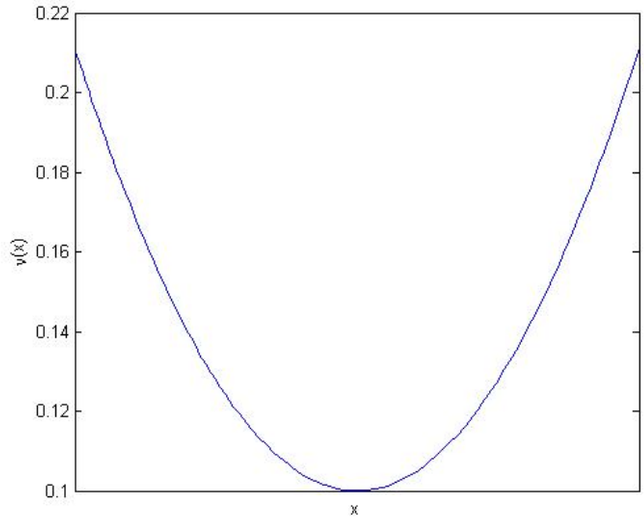


Figure 5.10: Graph of x versus $\nu(x)$.

Decision	Reward
$[2,0,0,0,0]$	4.5574
$[0,2,0,0,0]$	5.7668
$[0,0,2,0,0]$	6.1902
$[0,0,0,2,0]$	5.7448
$[0,0,0,0,2]$	4.5574

Table 5.7: Reward for hosts for the stable solutions with two hosts and five time points, with $C_R(x,t) = \nu(t)e^{x/2}$

the best choice to be $[0,0,2,0,0]$. Which is exactly as we would expect because this is when we set the hosts costs to be at its lowest.

However, if we now alter the number of *Good* nests. We expand this so now we have four hosts and ten possible starting time points. We get the outcomes for different numbers of *Good* nests as shown in Table 5.8.

Before we look at what these decisions mean individually we note the fact that the overall trend of these solutions is to get into one of the two best starting points (since we have an even number

Good Nests	Decision
1	[0,1,0,0,1,0,0,2,0,0]
2	[0,0,2,0,0,2,0,0,0,0]
3	[0,0,0,0,3,0,0,1,0,0]
4	[0,0,0,0,4,0,0,0,0,0]

Table 5.8: Decision by the host for increasing numbers of *Good* nests, with four hosts and ten starting time points with $C_R(x,t) = \nu(t)e^{x/2}$

of nests the cost of raising for the fifth and sixth time points is the same).

For the situation with only one *Good* nest, only three of the hosts are able to occupy these nests in the given season, since the *Bad* nests take away any potential help from breeding at the same time as other hosts. Thus the best choice for the three laying in the first nest must either be $[0, 0, 1, 0, 0, 1, 0, 0, 1, 0]$ or $[0, 1, 0, 0, 1, 0, 0, 1, 0, 0]$ since these two have one of them laying in the better starting points. The first one is better for the group as a whole. However if you are the host laying in the ninth time point you can actually better your reward by choosing time point two, making this solution unstable and can be disregarded. The value of $\nu(x)$ is the same at this time point but it has better relaying scenarios if it chooses to lay earlier. This results in the second of the above choices to be made. The fourth host must try and get the final *Good* nest.

The outcome for two *Good* nests is the best result for all the hosts and is uninvadeable. Its trying to occupy the two good nests in the best way possible. Because it can occupy two *Good* nests at once the hosts reap a bigger reward from laying in pairs. Like before taking time points three and six as opposed to two and five is to have the possibility of the host abandoning and getting a lower cost of raising should the worst happen.

As we stated the hosts are trying to get the most amount of hosts in the *Good* nests as possible. This can be seen in the case where we have three *Good* nests. Unlike the two previous examples the hosts choose time point five and eight to start their nests. The reason for this is that the three laying together will always be able to get one of the good nests if it starts again before time point eight. So choosing to start at five means it can abandon it immediately and start at the time point with the lowest cost of raising should it need to. The other one chooses to start laying at time point eight so it can get one of the good nests again.

The final one with all *Good* nests gives the expected result of them all starting in time point five. All can abandon and still get a *Good* nest with the lowest cost of raising.

Laying is better at the beginning of the breeding season

Suppose that we change our $\nu(x)$ to be just

$$\nu(t) = \left(\frac{t}{m+1}\right)^2 + 0.1,$$

a function so the cost of raising is less at the beginning of the season. The uniform $\nu(x)$ always gave a decision of $[m, 0, \ldots, 0]$ if there were m good nests. This case gives this decision no matter

the number of *Good* nests. This is because the difference in cost of raising overrules the detrimental effect of having a *Bad* nest. This value for $\nu(t)$ can be seen as a fast increasing function. This has been done only to emphasize the fact that these costs can overrule the difference in the number of *Good* and *Bad* nests. There will be a change over point where the difference in the number of *Good* and *Bad* nests has a larger effect than the cost of raising.

5.6 Discussion

We have developed a model of the interaction between hosts of a brood parasite. This is based upon the host having a set of discrete starting points with which to choose from and the knowledge of the number hosts and parasites that exist. This game uses the multiple parasite model of the previous section repeatedly with different values for λ, C_a and C_b, which are the variables most affected by the choices the hosts make. λ is affected because it is calculated by dividing the number of parasites by the number of hosts and C_a and C_b come from the possibility of restarting the nesting process.

We have seen that the factors affecting what kind of outcomes can be used are the amount of good nesting sites available, the number of parasites during the season and the raising costs during the season.

The decrease in good nesting sites makes the hosts diverge from the seemingly logical scenario that they should all lay at the same time. This is because the detrimental effect of having a bad nest outweighs any potential benefit from laying at the same time as other hosts. Clarke et al [5] have shown that sites with a good line of sight for the Cuckoo have a higher possibility of being parasitised. This would qualify as a *Bad* nest. Øien et al [35] showed that parasitism rates of Reed Warblers are related to the distance of the nest from the nearest tree. We thus adapt the model giving a ν value dependent on the distance that nest is from a tree. The number of parasites in a given time period could be described purely as the number of parasites around that season. It could also, for example, mean that there is just one parasite but it can parasitise more than one nest during a given time period. It could even be a combination of the two. As you would expect there is a fairly linear relationship between the number of parasites and the hosts reward. Many studies have shown parasitism rates stay the same throughout the breeding season, and we have assumed this in our examples. However a possible next step for this game could be to alter the number of parasites throughout the season.

The effect of altering the raising costs is related to the assumption that food and other provisions change in their abundance throughout the season or simply that the ease in obtaining them alters. The effect of changing this cost means the host will tend to raise when it is least costly to do so, although it must balance this with the effect of being parasitised.

These models have all used small numbers of hosts. This is to give a general feel of how the model works. In practice much larger models like those shown in Figure 5.7, with models of up to twenty or even more hosts, would relate to more realistic populations.

Chapter 6

Discussion

This project set out to model the interaction between brood parasites and their hosts, beginning with seeing what sort of information the host and parasite would logically and mathematically have to arrive at the overall decision. It was decided that the problem was one of incomplete information, and that the host would decide what it was to do based upon what it sees at different intervals. For instance, if there is only one egg in the nest or only one has hatched should the host raise this hatchling or not? If it is not its own egg, it could lose the rest of the season where it may be able to raise another brood, and also waste energy on something that is not going to be beneficial to itself. However the hosts choosing to destroy the nest could have the opposite effect. So it must weigh up the possibilities with the potential reward.

Modelling this game started by looking at this and allowing the host to lay *many* or *few* eggs, with the parasite adult and chick responding by destroying or not in kind (to *many*, *few* or *one*). Whilst this gave an insight into how the host could come to raise just one egg, the model was flawed in that it did not allow the host to differentiate between many and one less than many. Because of this, one of the host's main defences of egg ejection was difficult to implement. It did however give us some mixed solutions where we can have different decisions by the host and parasite that are not just raise or not raise but raise with a certain probability and the same for the parasite decision. Because of the way the decisions and the variables are set up in the later games we do not get these interesting solutions.

To improve upon this we allowed the host to lay any number of eggs again, with the parasite allowed to respond in turn. This brought in, even more, the effect of the probability of parasitism, ejection success (and in turn mimicry) and parasite raising costs, which have been shown to be

the main factors in parasitism. We conclude that the host and parasite in general lose out due to increasing parasitism rates. The reason for the host receiving a smaller reward is obvious, but the parasite's reward decreases as a result of increased defences by the host. The other factors had the expected reward (increase costs and get a smaller reward).

The multiple parasite model built upon the previous model by adding in the possibility of more than one parasite visiting the nest, which is something that happens in nature and makes this model more realistic. In this, the parasitism probability β became a Poisson process with rate λ. As before the increase in rate resulted in a decrease in host and parasite reward. However instead of the parasite's decrease in reward being a secondary effect of increased defences by the host (although this still does happen) it actually has a primary effect of having other parasites competing for the same nest.

The time-based model is built upon the multiple parasite model looking at the effect of multiple hosts laying together in a breeding season. The choice of starting time point and nesting site were put together to find the value of λ. This is then put into the multiple game to find the overall reward for that nesting site at that time point. Other factors such as raising costs changing during the season were also investgated, as were which variables could be altered to become a function of time. Then to see what difference they would make to the overall outcome. The interesting points come from looking at the number of *Good* nests that the hosts can use during the season. If there are infinitely many of these nests then the hosts will try and lay together to get the best overall reward. If there are not enough nests to accommodate the hosts then they will try and spread themselves out so that they reuse the *Good* nesting sites as much as they can. If this is still not a possibility then there will be hosts who lose out.

All of the models show that a small amount of parasitism does not affect the host greatly and is also beneficial to the parasite. However when the parasitism rate increases so does host defence against it. The patterns shown in the graphs for the parasitism rate are probably the most interesting factor of this project. The *jumps* shown in the β and λ graphs show a change in decision by either the host or parasite. This decision is usually the host raising less so it will lose out less when parasitism does occur.

This has been an important piece of work because it combines all aspects of the interaction. These models show that the interaction between host and parasite can be modeled using sequential decisions and by changing the parameters of the reward and cost functions we can find solutions

which match those of real world interactions such as those between Cuckoos and their hosts and Cowbirds and their hosts. It is also possible to use this model for all different kinds of brood parasites. This model could also be used in the future to predict changes in host and parasite behaviour if some natural change does occur, for example a decrease in nesting sites or an increase in natural predation or destruction. This work could also be used as a basis for modeling the interaction between host and parasite in other non-avian parasitism such as *Nomadinae* a type of bee referred to as a cuckoo bee [32]. These lay eggs in the spaces provisioned for other non-parasite bees and consume the larva. This could be seen as an interaction between one host chick and parasite chick but other impacts on the adjacent nesting sites could be included in a model. Obviously in different cases different aspects of the game would exist but the overall structure can still be applied.

Possible Additions

There are a number of possible additions which could be made to the game. These are other technical aspects or additional biological events which happen in nature but were not included in these models for reasons of complication.

It has been observed in some cases that parasites (mainly cowbirds) have laid more than one egg in a nest. This could be included in our game in a similar way to that of the host choosing the number of eggs to lay. This would then continue on to these parasites playing the game as a 'team'. Certain factors would have to be considered in setting up the game. First of all, do the parasites recognize each other? The parasite reward comes from the probability that it is the first to hatch. Having a number of parasite eggs being laid by multiple parasite adults complicates this probability. If you do choose to include the fact that parasites recognize each other (which seems sensible to do) then it will be necessary to *team up* these parasites. Teaming up causes further complications in that some of those laid could be destroyed naturally in Stage 3 or by other parasites in Stage 2. This means the already quite complex formula for calculating the first host to hatch becomes even more complex as more combinations have to be taken in account. In the chick game it will also be necessary to factor in that the first to hatch could have n 'team mates'. It was felt that this could be a good inclusion to the game but the added complications outweighed any potential benefit from adding this to the game at this stage. In fact choosing to have the parasites not recognizing each other and having the parasites only laying one egg per day gives the same

game as we have in the multiple parasite model, just having one parasite visit multiple times as opposed to many parasites visiting the nest.

Some papers have shown an action by the parasite (again usually Cowbirds) [48, 18] where it will return to a nest at a later date to check up on it and if the parasite egg has been destroyed then it will destroy the remaining host eggs (called a gangster mentality). This can be interpreted as an attempt to bully the host into raising the parasites alongside the host.

This could be included in the game by adding an extra section between Stages 4 and 5, where the parasite could return to the nest with a certain probability, assuming it can recognize its own egg it could destroy the nest if the parasite has been destroyed. The mathematical complications come when you begin to add multiple parasites visiting the nest. There are decisions which have to be made in the multiple parasite model. The main one is deciding whether it will destroy dependent on recognizing its own eggs or just eggs of its own species. This would probably only result in less ejection in Stage 4 so adding in this as a complication does not seem beneficial.

Something we could add into the time-based model is the possibility of hosts laying more than one brood in a season. This occurs in nature and would be something interesting to add in terms of the interaction between the hosts. One would have to see what happens if there are only a few *Good* nests and so on. However this would not affect the interaction between host and parasite a great deal. The aim of this project was to model sequentially the interaction between host and parasite and using the simplification that the host can only lay one brood. This however would be a worthwhile inclusion to the model and could be further developed in the future.

Bibliography

[1] A. J. Banks and T. E. Martin. Host activity and the risk of nest parasitism by brown-headed cowbirds. *Behavioral Ecology*, 12(1):31–40, 2001.

[2] D. R. Barber and T. E. Martin. Influence of alternate host densities on brown-headed cowbird parasitism rates in black-capped vireos. *Condor*, 99:595–604, 1997.

[3] M. d. L. Brooke, N. B. Davies, and D. G. Noble. Rapid decline of host defences in response to reduced cuckoo parasitism : behavioural flexibility of reed warblers in a changing world. *Proceedings of the Royal Society*, B(265):1277–1282, 1998.

[4] M. Broom, G. D. Ruxton, and R. M. Kilner. Host life history strategies and the evolution of chick-killing by brood parasitic offspring. *Behavioral Ecology*, 19:22–34, 2008.

[5] A. L. Clarke, I. J. Øien, M. Honza, A. Moksnes, and E. Roskaft. Factors affecting reed warbler risk of brood parasitism by the common cuckoo. *The Auk*, 118:534–538, 2001.

[6] N. B. Davies. Cuckoos, cowbirds and other cheats. *T. & A.D. Poyser*, 2000.

[7] N. B. Davies, M. D. L. Brooke, and A. Kacelnik. Recognition errors and probability of parasitism determine whether reed warblers should accept or reject mimetic cuckoo eggs. *Proceedings: Biological Sciences*, 263(1372):925–931, 1996.

[8] N. B. Davies, S. H. M. Butchart, T. A. Burke, N. Chaline, and I. R. K. Stewart. Reed warblers guard against cuckoos and cuckoldry. *Animal Behaviour*, 65(2):285–295, 2003.

[9] N. B. Davies and M. de L. Brooke. Cuckoos versus reed warblers: Adaptations and counter-adaptations. *Animal Behaviour*, 36(1):262–284, 1988.

[10] M. C. De Marsico and J. C. Reboreda. Differential reproductive success favours strong hostpreference in a highly specialized brood parasite. *Proceedings of the Royal Society*, 275:2499–2506, 2008.

[11] D. C. Dearborn. Video documentation of a brown-headed cowbird nestling ejecting an indigo bunting nestling from the nest. *Condor*, 98(3):645–649, 1996.

[12] R. M. Fraga. Interactions of the parasitic screaming and shiny cowbirds (*Molothrus rufoaxillaris and M. bonariensis*) with a shared host, the bay-winged cowbird (*M. badius*). *Parasitic Birds and Their Hosts (Ed. by S. I. Rothstein & S. K. Robinson)*, pages 173–193, 1998.

[13] A. J. Gaston. Brood parasitism by the pied crested cuckoo *Clamator jacobinus*. *Journal of Animal Ecology*, 45(2):331–348, 2003.

[14] C. A. Haas and K. H. Haas. Brood parasitism by brown-headed cowbirds on brown thrashers: frequency and rates of rejection. *Condor*, 100:535–540, 1998.

[15] M. D. Harrison and M. Broom. A game-theoretic model of interspecific brood parasitism with sequential decisions. *Journal of Theoretical Biology*, 256:504–517, 2009.

[16] J. P. Hoover. Multiple effects of brood parasitism reduce the reproductive success of prothonotary warblers, *Protonotaria citrea*. *Animal Behaviour*, 65(5):923–934, 2003.

[17] J. P. Hoover and M. J. Reetz. Brood parasitism increases provisioning rate, and reduces offspring recruitment and adult return rates, in a cowbird host. *Oecologia*, 149(1):165–73, 2006.

[18] J. P. Hoover and S. K. Robinson. Retaliatory mafia behavior by a parasitic cowbird favors host acceptance of parasitic eggs. *Proceedings of the National Academy of Sciences of the United States of America.*, 104:4479–4483, 2007.

[19] S. A. Hosoi and S. I. Rothstein. Nest desertion and cowbird parasitism: evidence for evolved responses and evolutionary. *Animal Behaviour*, 59:823–840, 2000.

[20] P. A. Johnsgard. *The Avian Brood Parasites: Deception at the Nest.* Oxford University Press US, 1997.

[21] R. M. Kilner. The evolution of virulence in brood parasites. *Ornithological Science*, 4:55–64, 2005.

[22] R. M. Kilner and N. B. Davies. How selfish is a cuckoo chick? *Animal Behaviour*, 58:787–808, 1999.

[23] R. M. Kilner, J. R. Madden, and M. E. Hauber. Brood parasitic cowbirds use host young to procure food. *Science*, 305:87, 2004.

[24] O. Krüger and N. B. Davies. The evolution of egg size in the brood parasitic cuckoos. *Behavioural Ecology*, 15:210–218, 2004.

[25] N. E. Langmore, S. Hunt, and R. M. Kilner. Escalation of a coevolutionary arms race through host rejection of brood parasitic young. *Nature*, 422:157–160, 2003.

[26] M. J. Lawes and T. R. Marthews. Reduced parasitism by retaliatory cuckoos selects for hosts that rear cuckoo nestlings. *Behavioral Ecology*, 14:757–770, 2003.

[27] J. C. Lorenzana and S. G. Sealy. Fitness costs and benefits of cowbird egg ejection by gray catbirds. *Behavioral Ecology*, 12(3):325–329, 2001.

[28] P. E. Lowther. Brown-headed cowbirds. *Birds of North America*, 47:1–24, 1995.

[29] J. Maruyama and H. Seno. Mathematical modelling for intra-specific brood-parasitism: coexistence between parasite and non-parasite. *Mathematical Biosciences*, 156(1):315–338, 1999.

[30] J. Maruyama and H. Seno. The optimal strategy for brood-parasitism: how many eggs should be laid in the host's nest?. *Mathematical Biosciences*, 161(1):43–63, 1999.

[31] C. M. McLaren, W. B. E., H. L. Gibbs, and S. G. Sealy. Genetic and temporal patterns of multiple parasitism by brown-headed cowbirds (*Molothrus ater*) on song sparrows (*Melospiza melodia*). *Canadian Journal of Zoology*, 81(2):281–286, 2003.

[32] C. D. Michener. *The Bees of the World*. Johns Hopkins University Press, 2000.

[33] C. Moskát. Common cuckoo parasitism in europe: behavioural adaptations, arms race and the role of metapopulations. *Ornithological Science*, 4:3–15, 2005.

[34] C. Moskát and M. Honza. Cuckoo (*Cuculus canorus*) parasitism and host's rejection behaviour in a heavily parasitized great reed warbler (*Acrocephalus arundinaceus*) population. *Ibis*, 144(4):614–622, 2002.

[35] I. J. Øien, M. Honza, A. Moksnes, and E. S. Roskaft. The risk of parasitism in relation to the distance from reed warbler nests to cuckoo perches. *Journal of Animal Ecology*, 65:147–153, 1996.

[36] C. P. Ortega. *Cowbirds and Other Brood Parasites*. University of Arizona Press, 1998.

[37] M. Pagel, A. P. Møller, and A. Pomiankowski. Reduced parasitism by retaliatory cuckoos selects for hosts that rear cuckoo nestlings. *Behavioral Ecology*, 9:566–572, 1998.

[38] R. B. Payne. The ecology of brood parasitism in birds. *Annual Review of Ecological Systems*, 8:1–28, 1977.

[39] R. B. Payne and L. L. Payne. Brood parasitism by cowbirds: Risks and effects on reproductive success and survival in indigo buntings. *Behavioral Ecology*, 9:364–373, 1998.

[40] R. Planque, N. F. Britton, N. R. Franks, and M. A. Peletier. The adaptiveness of defence strategies against cuckoo parasitism. *Bulletin of Mathematical Biology*, 64:1045–1068, 2002.

[41] A. Putaala and R. Hissa. Breeding dispersal and demography of wild and hand-reared grey partridges *Perdix Perdix*. *Wildlife Biology*, 4:137–145, 1998.

[42] M. Robert and G. Sorci. The evolution of obligate interspecific brood parasitism in birds. *Behavioral Ecology*, 12(2):128–133, 2001.

[43] M. A. Rodríguez-Gironés and A. Lotem. How to detect a cuckoo egg: a signal-detection theory model for recognition and learning. *American Naturalist*, 153:633–648, 1999.

[44] S. I. Rothstein. Evolutionary rates and host defenses against avian brood parasitism. *American Naturalist*, 109:161–176, 1975.

[45] K. A. Schmidt and C. J. Whelan. The relative impacts of nest predatiation of brood parasitism on season fecundity in songbirds. *Conservative Biology*, 13(1):46–57, 1999.

[46] S. G. Sealy. Burial of cowbird eggs by parasitized yellow warblers: an empirical and experimental study. *Animal Behaviour*, 49:877–889, 1995.

[47] M. R. Servedio and M. E. Hauber. To eject or to abandon? life history traits of hosts and parasites interact to influence the fitness payoffs of alternative anti-parasite strategies. *Journal of Evolutionary Biology*, 19(5):1585–1594, 2006.

[48] M. Soler, J. J. Soler, J. G. Martinez, and M. A. P. Magpie host manipulation by great spotted cuckoos: Evidence for an avian mafia? *Evolution*, 49:770–775, 1995.

[49] M. D. Sorenson and R. B. Payne. A molecular genetic analysis of cuckoo phylogeny. *Bird Families of the World: Cuckoos*, pages 68–94, 2005.

[50] B. G. Stokke, A. Moksnes, E. Skaft, G. Rudolfsen, and M. Honza. Rejection of artificial cuckoo (*Cuculus canorus*) eggs in relation to variation in egg appearance among reed warbler (*Acrocephalus scirpaceus*). *Biological Sciences*, 266(1427):1483–1493, 1999.

[51] F. Takasu. Why do all host species not show defense against avian brood parasitism: evolutionary lag or equilibrium? *American Naturalist*, 151(2):193–205, 1998.

[52] F. Takasu. A theoretical consideration on co-evolutionary interactions between avian brood parasites and their hosts. *Ornithological Science*, 4:65–72, 2005.

[53] F. Takasu, K. Kawasaki, H. Nakamura, J. E. Cohen, and N. Shigesada. Modeling the population dynamics of a cuckoohost association and the evolution of host defenses. *American Naturalist*, 142(5):819–839, 1993.

[54] J. J. Tewksbury, T. E. . Martin, S. J. Hejl, M. J. Kuehn, and J. W. Jenkins. Parental care of a cowbird host: caught between the costs of egg-removal and nest predation. *Proceedings of the Royal Society of London*, 269(2):423–429, 2002.

[55] E. van Damme. *Stability and Perfection of Nash Equilibria*. Springer Verlag, 1996.

[56] R. Winfree. Cuckoos, cowbirds and the persistence of brood parasitism. *Trends in Ecology & Evolution*, 14(9):338–343, 1999.

[57] I. Wyllie. *The Cuckoo*. Universe Books, 1981.

[58] A. Yamauchi. Theory of evolution of nest parasitism in birds. *American Naturalist*, 145(3):534–538, 1995.

[59] A. G. Zink. The evolution of intraspecific brood parasitism in birds and insects. *American Naturalist*, 155:395–405, 2000.

Appendix A

Single Interaction First Model

Stage1.m is the parent program with the others feeding off of it.

Stage1.m

```
function [HostOutcome,ParaOutcome,Eggs_Laid,ParaDecisionS2,a,b,Gamma] = Stage1(maxeggs
    ,Rh,Cr,Cl,Rp,Cda,Cdc,xi,beta,sigma,Ca,Cb,Ce);

for i = 1:maxeggs
    [HostStage2(i),ParaStage2(i),a,b,Gamma,alpha,ParaDecision(i)] = Stage2(i,Rh,Cr,Rp,
        Cda,Cdc,xi,beta,sigma,Ca,Cb,Ce);
    HostStage2(i) = HostStage2(i) - Cl(i);
end

[HostOutcome,Eggs_Laid] = max(HostStage2);

ParaOutcome = ParaStage2(Eggs_Laid);

ParaDecisionS2 = ParaDecision(Eggs_Laid);

%a = a(Eggs_Laid,:);

%b = b(Eggs_Laid,:);

%gamma = gamma(Eggs_Laid,:);
```

Stage2.m

```
function [HostStage2,ParaStage2,a,b,Gamma,alpha,ParaDecision] = Stage2(n,Rh,Cr,Rp,Cda,
    Cdc,xi,beta,sigma,Ca,Cb,Ce)

warning off all

ParaStage2 = -10000000000000;
HostStage2bTemp = zeros(1,n+1);
ParaStage2bTemp = zeros(1,n+1);
aTemp = zeros(1,n+1);
bTemp = zeros(1,n+1);
GammaTemp = zeros(n+2,n+2);

for z = 0:n

alphaTemp = zeros(1,n+1);

s = zeros(n+2,n+2);

s(1,1) = 1;

for j = 1:n+1
    s(j+1,j+1) = 0.99;
    for x = 0:j-1
       s(j+1,x+1) = 0.01/(j+1);
    end
end

for x = 0:n
    if z == x;
        Δ(x+1) = 1;
    else
        Δ(x+1) = 0;
    end
end
```

```
for i = 1:n+1

alpha_0 = 0;
alpha_notp = 0;
alpha_p = 0;



    alpha_0 = (1-beta)*s(n+1,i+1);

    for k = i:n
        alpha_notp = alpha_notp + Δ(k)*s(k+1,i+1)*(1-((i)/(k+1)));
    end
        alpha_notp = alpha_notp*beta;

    for k = i:n+1
        alpha_p = alpha_p + Δ(k)*s(k+1,i+1)*((i+1)/(k+1));
    end
        alpha_p = alpha_p*beta;

alphaTemp(i) = alpha_p/(alpha_p + alpha_notp + alpha_0);

TempI = i;

GammaTemp2 = zeros(1,i);

[HostStage2bTemp(i),ParaStage2bTemp(i),aTemp(i),bTemp(i),GammaTemp2] = Stage2b(TempI,
    Rh,Cr,Rp,Cdc,xi,alphaTemp(i),sigma,Ca,Cb,Ce);

GammaLength = length(GammaTemp2);

for l = GammaLength+1:n+2
    GammaTemp2(l) = 0;
end

GammaTemp(i,:) = GammaTemp2;

end
```

```
ParaStage2Temp = Δ(1)*(aTemp(1)*ParaStage2bTemp(1));
for j = 1:z
    ParaStage2Temp = ParaStage2Temp + s(z+1,j+1)*((j)/(z))*(aTemp(j+1)*ParaStage2bTemp
        (j+1) + bTemp(j+1)*(1-sigma)*ParaStage2bTemp(j+1));
end

ParaStage2Temp = ParaStage2Temp - Cda(n-z);

if ParaStage2Temp > ParaStage2
    ParaStage2 = ParaStage2Temp;
    HostStage2b = HostStage2bTemp;
    Δfinal = Δ;
    Gamma = GammaTemp(z+1,1:z+1);
    alpha = alphaTemp;
    ParaDecision = z;
end

    a = aTemp;
    b = bTemp;
end

HostStage2Para = 0;
HostStage2Para2 = 0;
HostStage2NoPara = 0;

for j = 1:n
for i = 0:ParaDecision
HostStage2Para = HostStage2Para + s(ParaDecision+1,i+1)*HostStage2b(i+1);
end
HostStage2Para = HostStage2Para*Δfinal(j+1);
HostStage2Para2 = HostStage2Para2 + HostStage2Para;
end
HostStage2Para = beta*HostStage2Para2;

for i = 1:n
```

```
HostStage2NoPara = HostStage2NoPara +  s(n+1,i+1)*HostStage2b(i);
end

HostStage2NoPara = (1—beta)*HostStage2NoPara;

HostStage2 = HostStage2Para + HostStage2NoPara;
```

Stage2b.m

```
function [HostStage2b,ParaStage2b,a,b,Gamma] = Stage2b(n,Rh,Cr,Rp,Cdc,xi,alpha,sigma,
    Ca,Cb,Ce)

[HostStage4_a,ParaStage4_a,Gamma_a] = Stage4(n,Rh,Cr,Rp,Cdc,1,0,xi,alpha,sigma,Ca,Cb);

[HostStage4_b,ParaStage4_b,Gamma_b] = Stage4(n,Rh,Cr,Rp,Cdc,0,1,xi,alpha,sigma,Ca,Cb);

HostStage4_b = HostStage4_b — Ce;

if HostStage4_a > HostStage4_b
    if HostStage4_a > Ca;
        a = 1;
        b = 0;
        HostStage2b = HostStage4_a;
        ParaStage2b = ParaStage4_a;
        Gamma = Gamma_a;
    else
        a = 0;
        b = 0;
        HostDecision2b = 3;
        HostStage2b = Ca;
        ParaStage2b = 0;
        Gamma = 0;
    end
else
    if HostStage4_b> Ca;
        a = 0;
```

```
        b = 1;
        HostStage2b = HostStage4_b;
        ParaStage2b = ParaStage4_b;
        Gamma = Gamma_b;
    else
        a = 0;
        b = 0;
        HostDecision2b = 3;
        HostStage2b = Ca;
        ParaStage2b = 0;
        Gamma = 0;
    end
end
```

Stage4.m

```
function [HostStage4,ParaStage4,Gamma] = Stage4(n,Rh,Cr,Rp,Cdc,a,b,xi,alpha,sigma,Ca,
    Cb)

t = zeros(n+2,n+2);t(1,1) = 1;
for j = 1:n+1
    t(j+1,j+1) = 0.99;
    for x = 0:j-1
        t(j+1,x+1) = 0.01/(j+1);
    end
end
gammavec = zeros(1,n+1);
gammavec2 = zeros(factorial(n+1),n+1);
        %temp = zeros(1,n+1);
        %flag = 0;
        %position(1,:) = temp;
        %for x = 1:factorial(n+1)-1
        %flag = 0;%pos = n;
        % while flag == 0
        %     if temp(pos+1) == pos
        %         for y = pos+1:n+1
```

```
%           temp(y) = 0;
%         end
%        if isequal(pos,0) == 0;
%             pos = pos-1;
%          end
%     else
%         flag = 1;
%         temp(pos+1) = temp(pos+1) + 1;
%      end
%   end
%gammavecnew(x+1,:) = temp;
%end%y = length(gammavecnew);
%for i = 1:y
% [HostStage4_2(i),ParaStage4_2(i)] = Stage5(n,Rh,Cr,Rp,Cdc,a,b,xi,alpha,sigma
    ,Ca,Cb,t,gammavecnew(i,:));
%end
%[ParaStage4_2,i] = max(ParaStage4_2);
%gammavecnew = gammavecnew(i,:);

[HostStage4Temp,ParaStage4Temp] = Stage5(n,Rh,Cr,Rp,Cdc,a,b,xi,alpha,sigma,Ca,
    Cb,t,gammavec);

for i = 1:n+1
    for j = n-i+1:-1:0
        gammavectemp = gammavec;
        gammavectemp(n-i+2) = j;
        [HostStage4Temp2,ParaStage4Temp2] = Stage5(n,Rh,Cr,Rp,Cdc,a,b,xi,alpha
            ,sigma,Ca,Cb,t,gammavectemp);
        if ParaStage4Temp2 > ParaStage4Temp
            ParaStage4Temp = ParaStage4Temp2;
            HostStage4Temp = HostStage4Temp2;
            gammavec = gammavectemp;
        end
    end
end

ParaStage4 = ParaStage4Temp;
```

```
        %[ParaStage4,i] = max(ParaStage4);
        HostStage4 = HostStage4Temp;
        %HostStage4 = HostStage4(i);
        %GammaVecTemp = gammavec;
        Gamma = gammavec;
        %for i = 1:y
        %GammaVecTemp(i,n+2) = ParaStage4Temp(i);
        %end
```

Stage5.m

```
function [HostStage5,ParaStage5] = Stage5(n,Rh,Cr,Rp,Cdc,a,b,xi,alpha,sigma,Ca,Cb,t,
    gammavec)

HostStage5 = zeros(1,n+2);
HostStage5NoPara = zeros(1,n+2);
HostStage5ParaA = zeros(1,n+2);
HostStage5ParaB = zeros(1,n+2);
Rho = zeros(1,n+2);
HostStage5NoPara(1) = 0;

for x = 0:n
    for y = 0:x
        if gammavec(x+1) == y;
            gamma(x+1,y+1) = 1;
        else
            gamma(x+1,y+1) = 0;
        end
    end
end

for i = 1:n+1
    HostStage5NoPara(i+1) = (1-alpha)*(a*t(n,i)*(Rh(i)-Cr(i)) + b*t(n,i+1)*(Rh(i)-Cr(i
        )));
end
```

```
for i = 1:n+1
    tempHostStage5ParaA = 0;
    for j = i:n+1
        tempHostStage5ParaA = tempHostStage5ParaA + t(n+2,j+1)*((j+1)/(n+2))*gamma(j,i
            )*(Rh(i-1)-Cr(i+xi-1));
    end
    HostStage5ParaA(i+1) = alpha*(a*(tempHostStage5ParaA + t(n+1,i+1)*(1-((x)/(n+1)))
        *(Rh(i)-Cr(i))));
end

for i = 1:n
    tempHostStage5ParaB = 0;
    for j = i:n
        tempHostStage5ParaB = tempHostStage5ParaA + t(n,j+1)*((j+1)/(n))*gamma(j,i)*(
            Rh(i-1)-Cr(i+xi-1));
    end
    HostStage5ParaB(i+1) = alpha*(b*(sigma*t(n,i) + (1-sigma)*(tempHostStage5ParaA + t
        (n,i+1)*(1-((x)/(n)))*(Rh(i)-Cr(i)))));
end

for i = 0:n+1
    HostStage5(i+1) = HostStage5NoPara(i+1) + HostStage5ParaA(i+1) + HostStage5ParaB(i
        +1) + (1-(a+b))*Ca;
end

for i = 0:n+1
    if HostStage5(i+1) < Cb
        Rho(i+1) = 0;
    else
        Rho(i+1) = 1;
    end
end

tempParaStage5 = 0;

for i = 0:n
    tempParaStage5 = 0;
    for j = 0:i
```

```
        tempParaStage5 = tempParaStage5 + gamma(i+1,j+1)*((Rho(j+2)*(Rp(j)))-Cdc(i-j))
            ;
    end
    ParaStage5(i+1) = t(n+1,i+1)*tempParaStage5;
end


HostStage5 = sum(HostStage5);
ParaStage5 = sum(ParaStage5);
```

Appendix B

Multiple Model

MultStage1.m is the parent program with the others feeding off of it.

MultStage1.m

```
function [HostOutcome,ParaOutcome,n_h,ParaDecision,HostDecision,Δ,chickabandon,
    adultabandon] = MultStage1(maxn,lambda,Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,C_a,C_b,
    tol)

HostDecision1 = zeros(maxn,maxn);

for n = 1:maxn
    [HostOutcome1(n),ParaOutcome1(n),ParaDecision1(n),HostDecision1(n,1:n+1),Δ1(n,1:n
        +1),chickabandon1(n),adultabandon1(n)] = MultStage2(n,lambda,Rh,Cr,Rp,Cda,Cdc,
        Ci,xi,sigma,Ce,C_a,C_b,tol);
    HostOutcome1(n) = HostOutcome1(n) - Cl(n);
end



[HostOutcome,n_h] = max(HostOutcome1);



ParaOutcome = ParaOutcome1(n_h);
```

```
HostDecision = HostDecision1(n_h,1:n_h);
ParaDecision = ParaDecision1(n_h);
Δ = Δ1(n_h,1:n_h);
chickabandon = chickabandon1(n_h);
adultabandon = adultabandon1(n_h);
```

MultStage2.m

```
function [HostOutcome,ParaOutcome,ParaDecision,HostDecision,Δ,chickabandon,
    adultabandon] = MultStage2(n,lambda,Rh,Cr,Rp,Cda,Cdc,Ci,xi,sigma,Ce,C_a,C_b,tol)

maxnp = 0;
prob = 1;

while prob > tol
    prob = (exp(-lambda)*(lambda^maxnp))/factorial(maxnp);
    maxnp = maxnp +1;
    mu(maxnp) = prob;
end

for i = 0:n
    s = zeros(n+2,n+2);
    s(1,1) = 1;
    for j = 1:n+1
    s(j+1,j+1) = 0.99;
    for x = 0:j-1
        s(j+1,x+1) = 0.01/(j);
    end
    end
end

ParaOutcome1 = zeros(n+1,1);
HostOutcome1 = zeros(n+1,1);
```

```
for ΔA = 1:n
Stage2Outcome2 = 0;
Stage2Outcome =0;

for n_p = 2:maxnp
    for x_p = 1:n_p
        for v_p = 1:n_p
            Stage2Outcome3 = 0;
            for z_p = 0:v_p-1
                Stage2Outcome4=0;
                for yy = 0:ΔA-1
                    Stage2Outcome4 = Stage2Outcome4 + s(ΔA+1,yy+1)*((z_p/(ΔA-yy))^(z_p
                        ));
                end
                Stage2Outcome4=Stage2Outcome4 + s(ΔA+1,ΔA+1);
                if n_p-v_p < x_p-z_p
                    Stage2Outcome3 = Stage2Outcome3;
                else
                    if x_p<z_p
                        Stage2Outcome3 = Stage2Outcome3 ;
                    else
                        if z_p<1
                        Stage2Outcome3 = Stage2Outcome3 ;
                        else
                        Stage2Outcome3 = Stage2Outcome3 + ((nchoosek(v_p-1,z_p-1)*
                            nchoosek(n_p-v_p,x_p-z_p))/nchoosek(n_p,x_p));
                        end
                    end
                end
                Stage2Outcome6 = Stage2Outcome3*Stage2Outcome4;
            end
            Stage2Outcome2 = Stage2Outcome2 + ((ΔA-1)/ΔA)^(n_p-v_p) * Stage2Outcome6;
        end
    Stage2Outcome2 = Stage2Outcome2*(1/n_p)*phi(n_p-1,x_p-1,ΔA);
    end
    Stage2Outcome5=0;
    for y = 0:ΔA
```

```
        Stage2Outcome5 = Stage2Outcome5 + s(ΔA+1,y+1)*(y/ΔA);
    end
    Stage2Outcome = Stage2Outcome + Stage2Outcome2*Stage2Outcome5*mu(n_p);
end

Stage2Outcome = Stage2Outcome+mu(1);

[HostOutcome2,HostDecision1(ΔA,:),FirstParaOutcome,OtherParaOutcome,Δ1(ΔA,:),
    chickabandon1(ΔA,:)] = MultStage4(n,ΔA,lambda,Rh,Cr,Rp,Cdc,Ci,xi,sigma,Ce,C_a,C_b,
    tol,maxnp);




for i = 0:ΔA
    ParaOutcome1(ΔA) = ParaOutcome1(ΔA) + (Stage2Outcome)*s(ΔA+1,i+1)*FirstParaOutcome
        (i+1) + (1-Stage2Outcome)*s(ΔA+1,i+1)*OtherParaOutcome(i+1) - Cda(n-ΔA);
end

for j = 0:ΔA
    HostOutcome1(ΔA) = HostOutcome1(ΔA) + s(ΔA+1,j+1)*HostOutcome2(j+1);
end



end

[ParaOutcome,ParaDecision] = max(ParaOutcome1);

HostOutcome = HostOutcome1(ParaDecision);




HostDecision = HostDecision1(ParaDecision,:);
Δ = Δ1(ParaDecision,:);
chickabandon2 = chickabandon1(ParaDecision,:);

chickabandon = 0;
```

```
for i = 1:ParaDecision+1
    chickabandon = chickabandon + s(ParaDecision+1,i)*chickabandon2(i);
end

adultabandon = 0;

for i = 0:ParaDecision
    if HostDecision(i+1) == 3
        adultabandon =  adultabandon + s(ParaDecision+1,i+1);
    end
end
```

MultStage4.m

```
function [HostOutcome,HostDecision,FirstParaOutcome,OtherParaOutcome,Δ,chickabandon] =
     MultStage4(n,ΔA,lambda,Rh,Cr,Rp,Cdc,Ci,xi,sigma,Ce,C_a,C_b,tol,maxnp)

prob = 1;


for y = 0:maxnp
    prob = (exp(-lambda)*(lambda^y))/factorial(y);
    mu(y+1) = prob;
end




alpha = zeros(n+1,n+1);

for n_p = 0:maxnp
    for y=0:n
        for y_p=0:y
            alpha(y+1,y_p+1,n_p+1) = alphacalc(y,y_p,n_p,ΔA,n);
        end
    end
end
```

```
alpha2=0;

for y = 0:maxnp
    alpha2 = alpha2 + mu(y+1) * alpha(:,:,y+1);
end




A = zeros(n+1,1);

for y = 1:n
        [A(y+1),APara(y+1),AOtherPara(y+1),AΔ(y+1),Aprobabandon(y+1)] = MultStage5(y,
            alpha2(y,:),Rh,Cr,Rp,Cdc,xi,C_b);
        A(y+1) = A(y+1) - Ci(y);
end

B1 = zeros(n+1,1);




for y = 1:n
    for y_p = 0:y-1
    B1Alpha = alpha2;
    [B12(y+1),B1Para(y+1),B1OtherPara(y+1),B1Δ(y+1),B1probabandon(y+1)] = MultStage5(y
        -y_p,B1Alpha(y,:),Rh,Cr,Rp,Cdc,xi,C_b);
    B1(y+1) = B1(y+1) + alpha(y+1,y_p+1)*B12(y+1);
    B1(y+1) = B1(y+1) - Ci(y);
    end
end

B2 = zeros(n+1,1);

probtest = 0;




for y = 1:n
```

```
    [prob,maxmu] = max(alpha(y,:));
    B2Alpha = alpha*(1-sigma);
    if maxmu < y
    [B2(y+1),B2Para(y+1),B2OtherPara(y+1),B2Δ(y+1),B2probabandon(y+1)] = MultStage5(y-
        maxmu,B2Alpha(y,:),Rh,Cr,Rp,Cdc,xi,C_b);
    B2(y+1) = B2(y+1) - Ci(y);
    else
    [B2(y+1),B2Para(y+1),B2OtherPara(y+1),B2Δ(y+1),B2probabandon(y+1)] = MultStage5(y-
        maxmu,B2Alpha(y,:),Rh,Cr,Rp,Cdc,xi,C_b);
    B2(y+1) = B2(y+1) - Ci(y);
    end
end

for y = 1:n
    B(y+1) = B1(y+1)*sigma + B2(y+1)*(1-sigma) - Ce;
end

B(1) = 0;

C = ones(n+1,1);

C = C*C_a;

B = B';

HostMatrix = [A,B,C];
FirstPara = [APara;(1-sigma)*B2Para;zeros(1,n+1)];
FirstPara = FirstPara';
OtherPara = [AOtherPara;(1-sigma)*B2OtherPara;zeros(1,n+1)];
OtherPara = OtherPara';

Δ1 = [AΔ;B2Δ;zeros(1,n+1)]';
chickabandon1 = [Aprobabandon;sigma*B1probabandon + (1-sigma)*B2probabandon;zeros(1,n
    +1)]';



for i = 0:n
    [HostOutcome(i+1),HostDecision(i+1)] = max(HostMatrix(i+1,:));
```

```
end

for i = 0:n
    FirstParaOutcome(i+1) = FirstPara(i+1,HostDecision(i+1));
end

for i = 0:n
    OtherParaOutcome(i+1) = OtherPara(i+1,HostDecision(i+1));
end

for i = 0:n
    Δ(i+1) = Δ1(i+1,HostDecision(i+1));
end

chickabandon = 0;

for i = 0:n
    chickabandon(i+1) = chickabandon1(i+1,HostDecision(i+1));
end
```

MultStage5.m

```
function [HostStage5,FirstPara5,OtherPara5,Δ,ProbAbandon] = MultStage5(m,alpha,Rh,Cr,
    Rp,Cdc,xi,C_b)

for i = 1:m
[HostStage5(i),FirstPara(i),OtherPara(i),rho,prob(i,:),HostOutcomeCheck(i,:),
    probabandon(i)] = MultStage7(m,alpha,i,Rh,Cr,Rp,Cdc,xi,C_b);
end



if m==0
FirstPara5 = 0;
HostStage5=0;
OtherPara5=0;
Δ=0;
```

```
HostOutcomeCheck = zeros(m,1);
ProbAbandon = 0;
prob = 0;

else

[FirstPara5,Δ] = max(FirstPara);

HostStage5 = HostStage5(Δ);
OtherPara5 = OtherPara(Δ);
prob = prob(Δ,:);
HostOutcomeCheck = HostOutcomeCheck(Δ,:);
ProbAbandon = probabandon(Δ);




Δ = Δ − 1;
end
```

MultStage7.m

```
function [HostStage5,FirstPara,OtherPara,rho,Prob,HostOutcomeCheck,probabandon] =
    MultStage7(m,alpha,Δ,Rh,Cr,Rp,Cdc,xi,C_b)



HostOutcome = zeros(m+1,1);
Prob = zeros(m+1,1);
rho = zeros(m,1);
for i = 0:m
    t = zeros(m+2,m+2);
    t(1,1) = 1;
    for j = 1:m+1
    t(j+1,j+1) = 0.99;
    for x = 0:j−1
        t(j+1,x+1) = 0.01/(j);
    end
```

```
    end
end

for i = 1:m
    if Δ == i
        Δvec(i) = 1;
    else
        Δvec(i) = 0;
    end
end

for i = 1:m
for m_p = 1:m
    HostOutcome5 = 0;
    Prob5 = 0;
    m_h = m—m_p;
    HostOutcome4 = 0;
    Prob4 = 0;
    for x = 1:m
        HostOutcome3 = 0;
        Prob3 = 0;
        for x_h = 0:x
            x_p = x—x_h;
            HostOutcome2 = 0;
            Prob2 = 0;
                HostOutcome1 = 0;
                Prob1 = 0;
                for y_h = 1:i
                    y_p = i—y_h;
                    if x_h≥y_h
                        if x_p ≥ y_p
                            HostOutcome1 = HostOutcome1 + ((nchoosek(x_h,y_h)*nchoosek
                                (x_p,y_p))/nchoosek(x,i))*(Rh(y_h)—Cr(y_h+y_p*xi));
                            Prob1 = Prob1 + ((nchoosek(x_h,y_h)*nchoosek(x_p,y_p))/
                                nchoosek(x,i));
                        else
                            HostOutcome1 = HostOutcome1 + (Rh(y_h)—Cr(y_h));
                            Prob1 = Prob1;
```

```
                    end
                else
                HostOutcome1 = HostOutcome1;
                Prob1 = Prob1;
                end
            end
            HostOutcome2= HostOutcome2 + HostOutcome1*t(x+1,i+1);
            Prob2 = Prob2 + Prob1*t(x+1,i+1);
        if m_h≥x_h
            if x−x_h−1>0
                if m_p ≥ x_p
                    HostOutcome3 = HostOutcome3 + HostOutcome2*((nchoosek(m_h,x_h)
                        *nchoosek(m_p−1,x_p−1))/nchoosek(m−1,x−1));
                    Prob3 = Prob3 + Prob2*((nchoosek(m_h,x_h)*nchoosek(m_p−1,x_p
                        −1))/nchoosek(m−1,x−1));
                else
                    HostOutcome3 = HostOutcome3;
                    Prob3 = Prob3;
                end
            else
                HostOutcome3 = HostOutcome3;
                Prob3 = Prob3;
            end
        else
        HostOutcome3 = HostOutcome3;
        Prob3 = Prob3;
        end
    end
    HostOutcome4 = HostOutcome4 + HostOutcome3*Δvec(x);
    Prob4 = Prob4 + Prob3*Δvec(x);
end
HostOutcome(i+1) = HostOutcome(i+1) + HostOutcome4*alpha(m_p);
Prob(i+1) = Prob(i+1) + Prob4*alpha(m_p);
end
end

HostOutcome5 = 0;
```

```
for y = 0:m
    HostOutcome(y+1) = HostOutcome(y+1) + alpha(1)*t(m+1,y+1)*(Rh(y)—Cr(y));
    Prob(y+1) = Prob(y+1) + alpha(1)*t(m+1,y+1);
end



HostOutcomeCheck = HostOutcome;

for i = 1:m
    if HostOutcome(i+1) > C_b
        rho(i+1) = 1;
        HostOutcome(i+1) = HostOutcome(i+1);
    else
        rho(i+1) = 0;
        HostOutcome(i+1) = C_b*t(m+1,i);
    end
end

probabandon = 0;

for i = 0:m
    probabandon =  probabandon + t(i+1,m+1)*(1—rho(i+1));
end




HostOutcome(1) = 0;

HostStage5 = sum(HostOutcome);

FirstPara = 0;

for y =1:Δ
    FirstPara = FirstPara + t(Δ,y)*((y)/(Δ))*rho(y+1)*Rp(y—1);
end
FirstPara = FirstPara — Cdc(m—Δ);

OtherPara = 0;
```

```
    OtherPara1 = 0;
 for y=1:Δ
     OtherPara1 = OtherPara1 + t(Δ,y)*((y)/(Δ))*rho(y+1)*Rp(y-1);
 end
     OtherPara = OtherPara1*((Δ-1)/m);
```

alphacalc.m

```
function alpha = alphacalc(y,y_p,n_p,x,n_h)

alpha = 0;

if y_p ==0
    if n_p == 0
    alpha = 1;
    else
    alpha = 0;
    end
end




for x_p = 1:max(y,n_p)
    if x_p≥y_p
        if x-x_p ≥ y-y_p
            alpha = alpha + phi(n_p,x_p,x)*((nchoosek(x_p,y_p)*nchoosek(x-x_p,y-y_p))/
                nchoosek(x,y));
        else
            alpha = alpha ;
        end
    else
        alpha=alpha;
    end
```

```
end



end
```

calcnum.m

```
function [position] = calcnum(n,m)

m = m-1;

temp = zeros(1,m+1);
temp(m+1) = n;
flag = 0;;
position(1,:) = temp;
x = 1;

while position(x,1) < n
flag = 0;
pos = m;
    while flag == 0
        if temp(pos+1) == n
           for y = pos+1:m+1
               temp(y) = 0;
           end
           if isequal(pos,0) == 0;
                pos = pos-1;
           end
        else
           flag = 1;
           temp(pos+1) = temp(pos+1) + 1;
        end
    end
if sum(temp) == n
position(x+1,:) = temp;
x = x+1;
```

```
end
end

y = 1;

p = length(position);
```

phi.m

```
function prob = phi(n,x,p)









if x>n
    prob = 0;
else



if n<0
    prob =1;
else

if x>0
    vectornum = nchoosek(n-1,x-1);
else
    vectornum = 1;
end

vector = ones(vectornum,n-x);
prob = 0;
flag1 =0;

for i = 2:vectornum
```

```
    flag1 =0;
    j = n-x;
    while flag1 == 0
        if vector(i-1,j) == x
        vector(i:vectornum,j-1) = vector(i-1,j-1)+1;
        vector(i:vectornum,j:n-x) = vector(i,j-1);
        flag = 1;
        else
        vector(i,j) = vector(i-1,j)+1;
        flag1 =1;
        end
        j = j-1;
    end
end

survive = 1;

for k = 1:x-1
    survive = (p-k)*survive;
end



for i = 1:vectornum
    prob = prob + survive*prod(vector(i,:));
end

prob = (1/(p^(n-1)))*prob;
end

end
```

positions.m

```
function [Positions] = positions(Choice)

m = length(Choice);
```

```
n = sum(Choice);


    Positions = zeros(1,n);
    CheckSum = 1;
    for j = 1:m
        if Choice(j) == 0
        else
        Positions(CheckSum:CheckSum+Choice(j)-1) = j;
        CheckSum = CheckSum + Choice(j);
        end
    end
```

Appendix C

Time Based Model

TimeBased.m is the parent program with the others feeding off of it.

TimeBased.m

```
function [BestOutcome,BestChoice,A,Outcome,BestChoice2,TotalOutcome] = TimeBased(n,m,
    t_1,t_2,t_3,parasites,maxn,Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,tol,goodnests,oknests
    )


A = calcnum(n,m);


sollength = length(A);



Outcome = zeros(sollength,m);
Outcome2 = 0;


t = t_1 + t_2 + t_3;




for i = 1:goodnests
   nestvalue(i) = 0.01;
end
```

```
for i = goodnests+1:goodnests+oknests
   nestvalue(i) = 0.05;
end

for i = goodnests+oknests+1:n
    nestvalue(i) =1;
end

for i = 1:n
lambda1(i) = (parasites/n)*(nestvalue(i));
end




for i = 1:n
    Cr = @(x)(((m-((m+1)/2))/(m+1))^2+0.1)*exp(x/2);
    [HostOutcome,ParaOutcome,n_h,ParaDecision,HostDecision,Δ] = MultStage1(maxn,
        lambda1(i),Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,0,0,tol);
    Outcome2 = Outcome2 + HostOutcome;
end

HostOutcome1 = Outcome2/n;

HostOutcome(1,1:n) = HostOutcome1;

Outcome(1,m) = HostOutcome1;

p = 1;



BestChoice2(p,:) = A(1,:);
TotalOutcome(p) = sum(HostOutcome(1,:));

for j = 2:sollength

CurrentChoice = A(j,:);

B = TimeBasedCalc(n,m,CurrentChoice,t_1+t_2+t_3);
```

```
[HostInfo1(:,:,j),NewOutcome,HostOutcome(j,:)] = TimeBasedCheck(A,CurrentChoice,
    Outcome,n,m,t_1,t_2,t_3,parasites,maxn,Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,tol,
    nestvalue,BestChoice2);




flag1 = 0;

BestChoice1 = CurrentChoice;




    CurrentPositions = positions(BestChoice1);

    for k = 1:n
        for l=1:m
            if CurrentPositions(k) == l
                Outcome(j,l) = Outcome(j,l) + HostOutcome(j,k);
            end
        end
    end

    for l = 1:m
        if BestChoice1(l)== 0
            Outcome(j,l) = 0;
        else
        Outcome(j,l) = Outcome(j,l)/BestChoice1(l);
        end
    end



    Okay = 0;
    NewChoice = BestChoice1;
```

```
    for i = 1:n
        if HostInfo1(i,2,j) >0.98
            NewChoice(CurrentPositions(i)) = NewChoice(CurrentPositions(i))−1;
            NewChoice(HostInfo1(i,1,j)) = NewChoice(HostInfo1(i,1,j)) +1;
            HostInfo1;
        else
            Okay = Okay+1;
        end
        if HostInfo1(i,4,j) >0.98

            NewChoice(CurrentPositions(i)) = NewChoice(CurrentPositions(i))−1;
            NewChoice(HostInfo1(i,3,j)) = NewChoice(HostInfo1(i,3,j)) +1;
            HostInfo1;
        else
            Okay = Okay+1;
        end
    end
        if Okay == 2*n
            p = p+1;
            BestChoice2(p,:) = BestChoice1;
            TotalOutcome(p) = sum(HostOutcome(j,:));
        end


end

[BestOutcome,ChoiceNum] = max(TotalOutcome);



BestChoice = BestChoice2(ChoiceNum,:);
HostInfo2 = HostInfo1(:,:,ChoiceNum);

HostInfo = HostInfo2;
```

TimeBasedCheck.m

```
function [HostInfo,NewOutcome,HostOutcome] = TimeBasedCheck(A,BestChoice,Outcome,n,m,
    t_1,t_2,t_3,parasites,maxn,Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,tol,nestvalue,
    BestChoice2)

aa = 1;

B = TimeBasedCalc(n,m,BestChoice,t_1+t_2+t_3);

AdultAbandonTo = m;
ChickAbandonTo = m;

for j = 1:m
        if BestChoice(j) == 0
            Outcome2 = 0;
        else
            bestsol1 = 0;
            bestsol2 = 0;
            testanswer = zeros(1,m);
            for k = j+1:m
                testsol = BestChoice;
                testsol(j) = testsol(j)-1;
                testsol(k) = testsol(k)+1;
                for ll = 1:length(BestChoice2(:,1))
                    if testsol == BestChoice2(ll,:)

                for l = 1:length(A)
                    if testsol == A(l,:)
                        testanswer(k) = Outcome(l,k);
                    end
                end
                if j+t_1-1 < m
                [bestsol1] = max(testanswer(j+t_1:m));
                for q = 1:m
                    if bestsol1 == testanswer(q)
                        AdultAbandonTo = q;
                    end
                end
                else
```

```
    bestsol1 = 0;
    AdultAbandonTo = 0;
    end
    if j+t_1+t_2-1 < m
    [bestsol2] = max(testanswer(j+t_1+t_2:m));
    if bestsol2 == testanswer(q)
        ChickAbandonTo = q;
    end
    else
    bestsol2 = 0;
    ChickAbandonTo = 0;
    end
        else
        AdultAbandonTo = AdultAbandonTo;
        ChickAbandonTo = ChickAbandonTo;
        end
    end

end
Outcome2 = 0;
a = 0;
q = a+1;
while a < BestChoice(j)
    flag1 = 0;
    r = 1;
    while flag1 == 0;
        if r == m+1;
            q = q+1;
            r = 1;
        else
        if B(q,r) == j;
            a = a+1;
            nest(a) = q;
            flag1 = 1;
            q = q+1;
        else
            r = r+1;
        end
```

```
            end
        end
    end
    for u = 1:n
         lambda(u) = (parasites/BestChoice(j))*(nestvalue(u));
    end
        for s = 1:BestChoice(j)

        Cr = @(x)(((j-((m+1)/2))/(m+1))^2+0.1)*exp(x/2)  ;

        [HostOutcome(aa),ParaOutcome,n_h,ParaDecision,HostDecision,Δ,
            chickabandon,adultabandon] = MultStage1(maxn,lambda(nest(s)),Rh,Cr,
            Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,bestsol1,bestsol2,tol);
        if chickabandon == 0
            ChickAbandonTo = 0;
        else
            if ChickAbandonTo == 0
                ChickAbandonTo = m;
            end
        end
        if adultabandon == 0
            AdultAbandonTo = 0;
        else
            if AdultAbandonTo == 0
                AdultAbandonTo = m;
            end
        end


        HostInfo(aa,:)=[AdultAbandonTo,adultabandon,ChickAbandonTo,
            chickabandon,bestsol1,bestsol2];
        Outcome2 = Outcome2 + HostOutcome(aa);
        aa = aa+1;
        end
    Outcome2 = Outcome2/BestChoice(j);
end
NewOutcome(aa,j) = Outcome2;
```

```
end
```

Appendix D

Other Code

Graphmaker.m

```
function [HostOutcome,FirstPara,OtherPara] = graphmaker(n,lambda,Rh,Cr,Rp,Cda,Cdc,Ci,
    Cl,xi,sigma,Ce,C_a,C_b,tol);



for i = 1:100
    i
    y(i) = 0.05*i;
    lambda = 0.05*i
    [HostOutcome(i),ParaOutcome(i),n_h,ParaDecision,HostDecision,Δ,chickabandon,
        adultabandon] = Stage1(n,lambda,Rh,Cr,Rp,Cda,Cdc,Ci,Cl,xi,sigma,Ce,C_a,C_b,tol)
        ;
end


hl1 = scatter(y,HostOutcome);
ylabel('Host Outcome')
xlabel('\lambda')
legend('Host')
ax1 = gca;
set(ax1)

ax2 = axes('Position',get(ax1,'Position'),...
         'YAxisLocation','right',...
```

```
         'Color','none',...
          'XColor','k','YColor','k');
hold on

hl2 = scatter(y,ParaOutcome,'x')
ylabel('Parasite Outcome')
legend('Parasite', 'Other Parasite')




%scatter(y
%scatter(y,ParaOutcome)
```

Variables.m

```
Rh = @(x) x;
Rp = @(x) exp(-x*0.1);
Cr = @(x) 0.25*exp(x/2);
Rp = @(x) (-((x-2.5))^2+ 50)/50;
Cda = @(x) 0.01*x;
Cdc = @(x) 0.01*x;
Cl = @(x) 0.01*x;
Ci = @(x) 0.01*x;
maxeggs = 10;
```